UNCLASSIFIED

AD NUMBER
AD832426
LIMITATION CHANGES
TO: **Approved for public release; distribution is unlimited.**
FROM: **Distribution authorized to U.S. Gov't. agencies and their contractors; Administrative/Operational Use; APR 1968. Other requests shall be referred to Air Force Weapons Lab., Kirtland AFB, NM 87117.**
AUTHORITY
AFWL ltr 30 Nov 1971

THIS PAGE IS UNCLASSIFIED

AD832426

AFWL-TR-67-131, VOL II

AFWL-TR-67-131, VOL II

NUCLEAR EXPLOSION INTERACTION STUDIES

Volume II

Two-Dimensional Code Development

J. R. Triplett et al.

Gulf General Atomic Incorporated
San Diego, California 92112

Contract No. F29601-67-C-0014

TECHNICAL REPORT NO. AFWL-TR-67-131, VOL II

April 1968

DDC
RECEIVED
MAY 22 1968
B

AIR FORCE WEAPONS LABORATORY
Air Force Systems Command
Kirtland Air Force Base
New Mexico

This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of AFWL (WLRT), Kirtland AFB, NM, 87117.

332

DISCLAIMER NOTICE

THIS DOCUMENT IS THE BEST QUALITY AVAILABLE.

COPY FURNISHED CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

BLANK PAGE

AFWL-TR-67-131, Vol II

ACCESSION for	
CFSTI	WHITE SECTION ☐
DDC	BUFF SECTION ☑
UNANNOUNCED	☐
JUSTIFICATION	
BY	
DISTRIBUTION/AVAILABILITY CODES	
DIST.	AVAIL. and/or SPECIAL
2	

AIR FORCE WEAPONS LABORATORY
Air Force Systems Command
Kirtland Air Force Base
New Mexico

When U. S. Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report is made available for study with the understanding that proprietary interests in and relating thereto will not be impaired. In case of apparent conflict or any other questions between the Government's rights and those of others, notify the Judge Advocate, Air Force Systems Command, Andrews Air Force Base, Washington, D. C. 20331.

DO NOT RETURN THIS COPY. RETAIN OR DESTROY.

NUCLEAR EXPLOSION INTERACTION STUDIES

Volume II

Two-Dimensional Code Development

J. R. Triplett et al.
Gulf General Atomic Incorporated
San Diego, California 92112

Contract No. F29601-67-C-0014

TECHNICAL REPORT NO. AFWL-TR-67-131, Vol II

This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of AFWL (WLRT), Kirtland AFB, NM, 87117. Distribution is limited because of the technology discussed in the report.

FOREWORD

This report was prepared by Gulf General Atomic Incorporated, San Diego, California, under Contract F29601-67-C-0014. The research was funded by DASA under Project 5710, Subtask 07.017, Program Element 6.16.46.01H, and by ARPA Order 313, Program Element 6.25.03.01R.

Inclusive dates of research were 29 September 1966 to 27 October 1967. The report was submitted 13 March 1968 by the Air Force Weapons Laboratory Project Officer, Major John Bode (WLRT).

This report is published in four volumes: Volume I, Laser Phenomenology (classified CONFIDENTIAL); Volume II, Two-Dimensional Code Development; Volume III, The OUTPUT Code; and Volume IV, Material Property Codes. The first volume contains a classified report on interaction of laser radiation with solid targets and a brief description of calculations done in conjunction with experiments at the Air Force Weapons Laboratory. The remaining three volumes contain reports of code development efforts in the areas of radiative transfer, hydrodynamics, radiative absorption coefficients, and equations of state.

The projects described in this report are for the most part in an incomplete state of development. This is due in part to the nature of the existing computer programs themselves, which continue in a state of development as long as they are in use, and in part to the time scale involved in bringing new programs to a state of capability for solving real problems.

Gulf General Atomic staff personnel responsible for the direction of the research include J. H. Alexander, R. Brightman, R. S. Englemore, B. E. Freeman, W. B. Lindley, L. Norris, J. T. Palmer, L. M. Schalit, J. R. Triplett, and Mrs. Chris Imes. Contractor's report number is GA-7764, Vol II.

The cooperation of Dr. P. V. Avizonis, Major J. Bode, Capt C. C. David, Major G. Spillman, and Lt L. Stoessel of AFWL is gratefully acknowledged.

Other documents produced under this contract are: GAMD-7592, " A Numerical Scheme for First-Order Compton Scattering," J. T. Palmer, December 13, 1966; GAMD-7846, "Difference Equations for Heat Flow in Two Dimensions," J. R. Triplett, March 2, 1967; GAMD-7879, "A Modified Characteristic Method for Radiative Transfer," J. R. Triplett, March 17, 1967; GAMD-7889, "R D C D. A FORTRAN Input Routine," J. H. Alexander, March 24, 1967; GAMD-8333, "Hydrodynamic Equations

for Multidimensional Problems," J. R. Triplett, October 24, 1967; GAMD-8379, "A Brief Study of the Thermodynamic Properties of Several Low Z Elements at Low Temperature," L. M. Schalit, November 22, 1967.

This technical report has been reviewed and is approved.

JOHN BODE
Major, USAF
Project Officer

TRUMAN L. FRANKLIN
Colonel, USAF
Chief, Theoretical Branch

CLAUDE K. STAMBAUGH
Colonel, USAF
Chief, Research Division

ABSTRACT

(Distribution Limitation Statement No. 2)

The HECTIC code is a two-dimensional Eulerian radiative hydrodynamics code designed for nuclear and laser phenomenology applications involving primarily vapor-phase materials, with provisions for heat conduction and vaporization of condensed matter. In this report, a formulation is presented for an improved method of solving the hydrodynamic equations. Solution of the radiative transfer equations by the long-characteristic method is discussed, and computer codes utilizing this approach are presented. The short-characteristic method is also discussed. The nonequilibrium diffusion approximation in two dimensions is considered, and a study of algorithms for solving the implicit difference equations which arise is reported. Experimental codes utilizing the nonequilibrium diffusion and short-characteristic methods are presented.

CONTENTS

FIGURES

TABLES

SECTION I

HYDRODYNAMIC EQUATIONS FOR MULTIDIMENSIONAL PROBLEMS

The formulation of a computer method for solving the equations governing the flow of a compressible viscous fluid is presented in this section. The aims of the section are essentially to summarize and generalize currently used methods for treating multidimensional fluid flow, and to write down a recipe for computer code development.

The equations are introduced in a more general form than usual, namely for an arbitrary coordinate system. The special case of cylindrical coordinates is then described, and finally the problem is restricted to the axially symmetric case. Specific developments beyond the material in references 1, 2, and 3 include: (1) the representation of the viscous stress components in cylindrical coordinates; (2) the detailed specification of the Lax-Wendroff-Burstein pseudoviscosity terms in cylindrical geometry; (3) a formulation of the two-step Lax-Wendroff difference equations for cylindrical geometry which differs somewhat from that given, for example, by Burstein (Ref. 3); (4) a number of alternative formulations of the differential equations which may prove helpful in the treatment of specific problems; and (5) a suggested special device for adjusting the coefficient of the pseudoviscosity terms.

GLOSSARY

A	Arbitrary second-rank tensor field (Eq. (30)); coefficient of κ^2 in Eq. (100)
a	Arbitrary vector field (Eqs. (28), (29))
a_o, a_1, a_2	Coefficients of the expansion of Q in powers of J (Eqs. (68), (69), et seq.)
B	Coefficient of κ in Eq. (100)
C	Constant term in Eq. (100)
c	Adiabatic sound speed (Eq. (16) et seq.)
D/Dt	Lagrangian time derivative operator (Eq. (10) et seq.)
E	Specific internal energy, ergs/gram (Eq. (4) et seq.)
e	Total fluid energy density, ergs/cm^3 (Eqs. (3), (4), et seq.)
F	Terms whose r-derivatives appear in the flow equations (Eqs. (39), (40), (47), (48), (88), et seq.), or terms whose divergence appears in the flow equations (Eqs. (55 through (81))
G	Terms whose y-derivatives appear in the flow equations (Eqs. (34), (42), (47), (48), et seg.)
g	The metric tensor (or, without indices, its determinant). Values for cylindrical coordinates are given in Eqs. (32), (33), (34).
H	Terms (arising from the use of non-Cartesian coordinates) which are undifferentiated in the flow equations (Eqs. (39), (43), (47), (48), et seq.)
I	Terms whose θ-derivatives appear in the flow equations (Eqs. (39), (41)); the unit matrix (Eqs. (64) through (82))

i	A spatial index; after Eq. (85), the radial mesh index
J	The Jacobian of F with respect to U (Eq. (57), et seq.)
j	A spatial index; after Eq. (86), the axial mesh index
K	Specific kinetic energy, ergs/gram (Eqs. (4), (19), (24))
k	A spatial index
m	A spatial index (Eq. (30)); momentum density (Eqs. (56) through (59), (83), (84)); radial momentum density (Eqs. (50) through (54), (99), (100))
n	Axial momentum density (Eqs. (50) through (54), (99), (100)); time index (Eq. (73), et seq.)
P	Thermodynamic pressure, ergs/cm^3 (Eqs. (3), (6), et seq.)
Q	The Lax-Wendroff pseudoviscosity matrix (Eqs. (64), (68)); the Lapidus pseudoviscosity matrix (Eq. (82))
q	With superscript, total heat flux component (Eq. (3), et seq.)
q_1, q_2, q_3	Eigenvalues of Q (Eqs. (64), (65), et seq.)
r	Radial coordinate, cm
S	Specific entropy (Eq. (13), et seq.)
T	Temperature (Eq. (12), et seq.)
t	Time, sec
U	Densities of mass, momentum, and total energy (Eqs. (39), (40), (47), (48), et seq.)
u	With superscript, the fluid velocity (superscript is suppressed in Eqs. (64) through (69)); elsewhere, without superscript, the radial component of fluid velocity (Eq. (37), et seq.)
V	Terms involving viscous stress whose divergence appears in the flow equations (Eq. (83))
v	Axial component of fluid velocity (Eq. (37), et seq.)
w	Aximuthal component of fluid velocity (Eq. (37), etseq.)

x	Arbitrary spatial coordinate, cm
y	Axial coordinate, cm
α, β	Arbitrary unit vector fields (Eqs. (26), (27)); spatial indices (Eqs. (83), (84))
Γ	Derivative of pressure with respect to internal energy density at constant volume, dimensionless (Eq. (17), et seq.); with three indices, a Christoffel symbol (Eqs. (28) through (36))
Δ, ∇	Forward and backward difference operators
δ	Kronecker symbol (Eqs. (31), (57), (62)); central difference operator (Eq. (85), et seq.)
θ	Azimuthal coordinate, dimensionless
κ	Coefficient of pseudoviscosity, dimensionless (Eq. (65), et seq.)
μ	Shear viscosity, gm/(cm sec) (Eq. (4), et seq.)
μ_B	Bulk viscosity, gm/(cm sec) (Eq. (4), et seq.)
ξ	Basic mesh interval (Eq. (65), et seq.)
τ	Viscous stress (Eqs. (2), (5), et seq.)
ρ	Mass density, gm/cm^3
Φ	Terms involving pseudoviscosity whose divergence appears in the flow equations (Eqs. (66), (67))
ϕ	Thermal dissipation rate, ergs/(cm^3 sec) (Eq. (12) et seq.)
ψ	Viscous dissipation rate, ergs/(cm^3 sec) (Eqs. (9), (11), et seq.)

DIFFERENTIAL EQUATIONS

The following equations express conservation of mass, momentum, and total energy for a viscous fluid in a Euclidean space:

$$\frac{\partial \rho}{\partial t} + \partial_j(\rho u^j) = 0 \tag{1}$$

$$\frac{\partial}{\partial t}(\rho u^i) + \partial_j(\rho u^i u^j + g^{ij}P - \tau^{ij}) = 0 \tag{2}$$

$$\frac{\partial e}{\partial t} + \partial_j\,[u^j(e + P) - g_{ik}\tau^{kj}u^i + q^j] = 0 \tag{3}$$

where ρ is mass density, e is total energy density, P is thermodynamic pressure, u^i is fluid velocity, q^i is total heat flux, g^{ij} is the metric, and τ^{ij} is the residual viscous stress. It is convenient also to introduce the specific internal energy E, the shear viscosity μ, and the bulk viscosity μ_B, in terms of which one writes the energy relation

$$e = \rho(E + 1/2g_{ij}\, u^i u^j) = \rho(E + K) \tag{4}$$

the constitutive relation

$$\tau^{ij} = \mu(g^{ik}u^j{}_{,k} + g^{jk}u^i{}_{,k}) + (\mu_B - 2/3\,\mu)g^{ij}u^k{}_{,k} \tag{5}$$

and the state equation

$$P = P(\rho, E) \tag{6}$$

The symbols ∂_j and $,j$ are alternative notations for covariant differentiation with respect to the coordinate x^j.

Alternative forms of these equations are as follows:

$$\frac{D\rho}{Dt} + \rho u^j{}_{,j} = 0 \tag{7}$$

$$\rho\,\frac{Du^i}{Dt} + \partial_j(g^{ij}P - \tau^{ij}) = 0 \tag{8}$$

$$\rho\,\frac{DE}{Dt} + Pu^j{}_{,j} - \Psi + q^j{}_{,j} = 0 \tag{9}$$

where D/Dt is the Lagrangian time derivative operator

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + u^j\partial_j \tag{10}$$

and the quantity

$$\Psi = g_{ik}\tau^{kj}u^{i}_{,j} = 2/3\,\mu[(u^{1}_{,1} - u^{2}_{,2})^{2} + (u^{2}_{,2} - u^{3}_{,3})^{2} + (u^{3}_{,3} - u^{1}_{,1})^{2}]$$

$$+ \mu[(u^{1}_{,2} + u^{2}_{,1})^{2} + (u^{2}_{,3} + u^{3}_{,2})^{2} + (u^{3}_{,1} + u^{1}_{,3})^{2}]$$

$$+ \mu_{B}(u^{j}_{,j})^{2} \tag{11}$$

is the viscous dissipation rate. With the int oduction of specific entropy S and temperature $T = (\partial E/\partial S)_{\rho}$, the thermal dissipation rate is

$$\phi = -\frac{1}{T} q^{i} \frac{\partial T}{\partial x^{i}} \tag{12}$$

The specific entropy S may be divided into reversible and irreversible parts S_{rev} and S_{irr}, such that

$$\rho T \frac{DS_{rev}}{Dt} = -q^{j}_{,j} - \phi \tag{13}$$

$$\rho T \frac{DS_{irr}}{Dt} = \Psi + \phi \tag{14}$$

The total heating rate is the sum of these:

$$\rho T \frac{DS}{Dt} = \rho \frac{DE}{Dt} + Pu^{j}_{,j} = \Psi - q^{j}_{,j} \tag{15}$$

Let the adiabatic sound speed be denoted by c, where

$$c^{2} = \left.\frac{\partial P}{\partial \rho}\right|_{S} \tag{16}$$

and a parameter Γ be defined by

$$\Gamma = \frac{1}{\rho}\left.\frac{\partial P}{\partial E}\right|_{\rho} = \frac{1}{\rho T}\left.\frac{\partial P}{\partial S}\right|_{\rho} \tag{17}$$

Then the rate equation for P is

$$\frac{DP}{Dt} + \rho c^{2} u^{j}_{,j} = \Gamma(\Psi - q^{j}_{,j}) \tag{18}$$

Finally, the rate equation for specific kinetic energy K is

$$\rho \frac{DK}{Dt} + u^j P_{,j} - g_{ij} u^j \tau^{ik}_{,k} = 0 \tag{19}$$

Modified forms of Eqs. (7), (9), (15), (18), and (19) may be written for the logarithms of the positive definite scalars ρ, E, S, P, and K, respectively:

$$\frac{D}{Dt} \log \rho + u^j_{,j} = 0 \tag{20}$$

$$\frac{D}{Dt} \log E + \frac{1}{\rho E} (P u^j_{,j} - \Psi + q^j_{,j}) = 0 \tag{21}$$

$$\frac{D}{Dt} \log S = \frac{1}{\rho T S} (\Psi - q^j_{,j}) \tag{22}$$

$$\frac{D}{Dt} \log P + \frac{\rho c^2}{P} u^j_{,j} = \frac{\Gamma}{P} (\Psi - q^j_{,j}) \tag{23}$$

$$\frac{D}{Dt} \log K + \frac{1}{\rho K} (u^j P_{,j} - g_{ij} u^j \tau^{ik}_{,k}) = 0 \tag{24}$$

The characteristic equations are the following combinations of Eqs. (7), (8), and (18):

$$\frac{DP}{Dt} - c^2 \frac{D\rho}{Dt} = \Gamma(\Psi - q^j_{,j}) \tag{25}$$

$$\alpha_i \left[\frac{Du^i}{Dt} + \frac{1}{\rho} \partial_j (g^{ij} P - \tau^{ij}) \right] = 0 \tag{26}$$

$$\rho c \left[\beta_i \frac{D}{Dt} + c \partial_i \right] u^i + \left[\frac{D}{Dt} + c \beta_i g^{ij} \partial_j \right] P = c \beta_i \tau^{ij}_{,j} + \Gamma(\Psi - q^j_{,j}) \tag{27}$$

where α_i and β_i are arbitrary spatial unit vector fields.

In the remainder of this report, the solution of Eqs. (1) through (6) will be discussed.

CYLINDRICAL COORDINATES

The following formulae of tensor analysis are needed:

$$a^i_{,k} = \frac{\partial}{\partial x^k} a^i + \Gamma^i_{jk} a^j \tag{28}$$

$$a^k_{,k} = \frac{\partial}{\partial x^k} a^k + \Gamma^k_{jk} a^j = \frac{1}{\sqrt{g}} \frac{\partial}{\partial x^k} \sqrt{g}\, a^k \tag{29}$$

$$A^{ij}_{,j} = \frac{\partial}{\partial x^j} A^{ij} + \Gamma^j_{mj} A^{im} + \Gamma^i_{mj} A^{mj} = \frac{1}{\sqrt{g}} \frac{\partial}{\partial x^j} \sqrt{g}\, A^{ij} + \Gamma^i_{mj} A^{mj} \tag{30}$$

where g is the metric determinant and Γ^i_{jk} is a Euclidean Christoffel symbol. The following identities are also noted:

$$g_{ij} g^{jk} = \delta^k_i$$

$$g^{ij}_{,k} = 0 \tag{31}$$

$$\delta^i_{j,k} = 0$$

For cylindrical coordinates, let $x^1 = r$, $x^2 = \theta$, $x^3 = y$. Then $\sqrt{g} = r$, and the nonvanishing components of g^{ij} and Γ^i_{jk} are

$$g_{11} = g^{11} = g_{33} = g^{33} = 1 \tag{32}$$

$$g_{22} = r^2 \tag{33}$$

$$g^{22} = \frac{1}{r^2} \tag{34}$$

$$\Gamma^1_{22} = -r \tag{35}$$

$$\Gamma^2_{12} = \Gamma^2_{21} = \frac{1}{r} \tag{36}$$

Let the velocity components be denoted by $u^1 = u$, $u^2 = w$, $u^3 = v$. The

velocity gradient components and the velocity divergence are then

$$
\begin{aligned}
u^1_{,1} &= \frac{\partial u}{\partial r} & u^2_{,1} &= \frac{\partial w}{\partial r} + \frac{w}{r} & u^3_{,1} &= \frac{\partial v}{\partial r} \\
u^1_{,2} &= \frac{\partial u}{\partial \theta} - rw & u^2_{,2} &= \frac{\partial w}{\partial \theta} + \frac{u}{r} & u^3_{,2} &= \frac{\partial v}{\partial \theta} \\
u^1_{,3} &= \frac{\partial u}{\partial y} & u^2_{,3} &= \frac{\partial w}{\partial y} & u^3_{,3} &= \frac{\partial v}{\partial y}
\end{aligned}
\tag{37}
$$

$$
u^k_{,k} = \frac{\partial u}{\partial r} + \frac{u}{r} + \frac{\partial w}{\partial \theta} + \frac{\partial v}{\partial y}
$$

The components of the residual stress are

$$
\begin{aligned}
\tau^{11} &= 2\mu \frac{\partial u}{\partial r} + \left(\mu_B - \frac{2}{3}\mu\right) u^k_{,k} \\
\tau^{22} &= r^{-2}\left[2\mu\left(\frac{\partial w}{\partial_\theta} + \frac{u}{r}\right) + \left(\mu_B - \frac{2}{3}\mu\right) u^k_{,k}\right] \\
\tau^{33} &= 2\mu \frac{\partial v}{\partial y} + \left(\mu_B - \frac{2}{3}\mu\right) u^k_{,k} \\
\tau^{12} &= \tau^{21} = \mu\left(\frac{\partial w}{\partial r} + r^{-2}\frac{\partial u}{\partial \theta}\right) \\
\tau^{13} &= \tau^{31} = \mu\left(\frac{\partial v}{\partial r} + \frac{\partial u}{\partial y}\right) \\
\tau^{23} &= \tau^{32} = \mu\left(r^{-2}\frac{\partial v}{\partial \theta} + \frac{\partial w}{\partial y}\right)
\end{aligned}
\tag{38}
$$

The conservation equations, (1), (2), and (3), may now be written in matrix form:

$$
\frac{\partial U}{\partial t} + \frac{\partial F}{\partial r} + \frac{\partial I}{\partial \theta} + \frac{\partial G}{\partial y} + H = 0 \tag{39}
$$

where

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho w \\ \rho v \\ e \end{bmatrix} \qquad F = \begin{bmatrix} \rho u \\ \rho u^2 + P - \tau^{11} \\ \rho uw - \tau^{21} \\ \rho vu - \tau^{31} \\ u(e + P - \tau^{11}) - r^2\tau^{21}w - \tau^{31}v + q^1 \end{bmatrix} \tag{40}$$

$$I = \begin{bmatrix} \rho w \\ \rho uw - \tau^{12} \\ \rho w^2 + r^{-2}P - \tau^{22} \\ \rho vw - \tau^{32} \\ w(e + P - r^2\tau^{22}) - \tau^{12}u - \tau^{32}v + q^2 \end{bmatrix} \tag{41}$$

$$G = \begin{bmatrix} \rho v \\ \rho uv - \tau^{13} \\ \rho wv - \tau^{23} \\ \rho v^2 + P - \tau^{33} \\ v(e + P - \tau^{33}) \cdot \tau^{13}u - \tau^{23}w + q^3 \end{bmatrix} \tag{42}$$

$$H = \frac{1}{r}\begin{bmatrix} \rho u \\ \rho u^2 - \tau^{11} - \rho r^2 w^2 + r^2\tau^{22} \\ 3(\rho uw - \tau^{12}) \\ \rho vu - \tau^{31} \\ u(e + P - \tau^{11}) - r^2\tau^{21}w - \tau^{31}v + q^1 \end{bmatrix} \tag{43}$$

For axially symmetric problems without rotation,

$$w = \frac{\partial u}{\partial \theta} = \frac{\partial w}{\partial \theta} = \frac{\partial v}{\partial \theta} = \frac{\partial P}{\partial \theta} = 0 \tag{44}$$

The equations then simplify to

$$u^1_{,1} = \frac{\partial u}{\partial r} \qquad u^3_{,1} = \frac{\partial v}{\partial r}$$

$$u^2_{,2} = \frac{u}{r} \qquad u^1_{,3} = \frac{\partial u}{\partial y} \tag{45}$$

$$u^3_{,3} = \frac{\partial v}{\partial y} \qquad u^k_{,k} = \frac{\partial u}{\partial r} + \frac{u}{r} + \frac{\partial v}{\partial y}$$

and

$$\tau^{11} = 2\mu \frac{\partial u}{\partial r} + \left(\mu_B - \frac{2}{3}\mu\right) u^k_{,k}$$

$$\tau^{22} = r^{-2}\left[\frac{2\mu u}{r} + \left(\mu_B - \frac{2}{3}\mu\right)u^k_{,k}\right]$$

$$\tau^{33} = 2\mu \frac{\partial v}{\partial y} + \left(\mu_B - \frac{2}{3}\mu\ u^k_{,k}\right) \tag{46}$$

$$\tau^{13} = \tau^{31} = \mu\left(\frac{\partial v}{\partial r} + \frac{\partial u}{\partial y}\right)$$

and

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial r} + \frac{\partial G}{\partial y} + H = 0 \tag{47}$$

with

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{bmatrix} \qquad F = \begin{bmatrix} \rho u \\ \rho u^2 + P - \tau^{11} \\ \rho vu - \tau^{31} \\ u(e + P - \tau^{11}) - \tau^{31} v + q^1 \end{bmatrix}$$

$$G = \begin{bmatrix} \rho v \\ \rho uv - \tau^{13} \\ \rho v^2 + P - \tau^{33} \\ v(e + P - \tau^{33}) - \tau^{13} u + q^3 \end{bmatrix} \qquad H = \frac{1}{r}\begin{bmatrix} \rho u \\ \rho u^2 - \tau^{11} + r^2 \tau^{22} \\ \rho vu - \tau^{31} \\ u(e + P - \tau^{11}) - \tau^{31} v + q^1 \end{bmatrix} \tag{48}$$

From Eqs. (16) and (17),

$$\left.\frac{\partial P}{\partial E}\right|_{\rho} = \rho\Gamma \qquad \left.\frac{\partial P}{\partial \rho}\right|_{E} = c^2 - \Gamma\,\frac{I}{\rho} \tag{49}$$

Let $m = \rho u$ and $n = \rho v$; then, from Eq. (4),

$$E = \frac{e}{\rho} - \frac{m^2 + n^2}{2\rho^2} \tag{50}$$

Therefore,

$$\left.\frac{\partial P}{\partial \rho}\right|_{m,\,n,\,e} = \left.\frac{\partial P}{\partial \rho}\right|_{E} + \left.\frac{\partial P}{\partial E}\right|_{\rho} \left.\frac{\partial E}{\partial \rho}\right|_{m,\,n,\,e}$$

$$= c^2 - \frac{\Gamma}{\rho}(P + e) + \Gamma(u^2 + v^2) \tag{51}$$

$$\left.\frac{\partial P}{\partial m}\right|_{\rho,\,n,\,e} = -\Gamma u \tag{52}$$

$$\left.\frac{\partial P}{\partial n}\right|_{\rho,\,m,\,e} = -\Gamma v \tag{53}$$

$$\left.\frac{\partial P}{\partial e}\right|_{\rho,\,m,\,n} = \Gamma \tag{54}$$

JACOBIAN AND PSEUDOVISCOSITY MATRICES

Let Eqs. (1), (2), and (3), with viscous stress $\tau^{ij} = 0$ and source $q^j = 0$, be written as

$$\frac{\partial U}{\partial t} + F^j_{\ ,j} = 0 \tag{55}$$

$$U = \begin{bmatrix} \rho \\ [m^i] \\ e \end{bmatrix} \qquad F^j = \begin{bmatrix} m^j \\ \left[\dfrac{m^i m^j}{\rho} + g^{ij}P\right] \\ \dfrac{m^j}{\rho}(e + P) \end{bmatrix} \tag{56}$$

The Jacobian matrix, defined as $J^j = \partial F^j / \partial U$, is

$$J^j = \begin{bmatrix} 0 & [\delta^j_k] & 0 \\ \left[-\frac{m^i m^j}{\rho^2} + g^{ij} \frac{\partial P}{\partial \rho} \right] & \left[\frac{m^i}{\rho} \delta^j_k + \frac{m^j}{\rho} \delta^i_k + g^{ij} \frac{\partial P}{\partial m^k} \right] & \left[g^{ij} \frac{\partial P}{\partial e} \right] \\ -\frac{m^j}{\rho^2}(e + P) + \frac{m^j}{\rho} \frac{\partial P}{\partial \rho} & \left[\frac{e + P}{\rho} \delta^j_k + \frac{m^j}{\rho} \frac{\partial P}{\partial m^k} \right] & \frac{m^j}{\rho} \left(1 + \frac{\partial P}{\partial e} \right) \end{bmatrix} \tag{57}$$

where, corresponding to Eqs. (51) through (54),

$$\frac{\partial P}{\partial \rho} = c^2 - \frac{\Gamma}{\rho}(P + e) + \frac{\Gamma}{\rho^2} g_{ij} m^i m^j = c^2 - \Gamma \frac{e + P}{\rho} + \Gamma g_{ij} u^i u^j \tag{58}$$

$$\frac{\partial P}{\partial m^k} = -\frac{\Gamma}{\rho} g_{ki} m^i = -\Gamma g_{ki} u^i \tag{59}$$

$$\frac{\partial P}{\partial e} = \Gamma \tag{60}$$

The matrix for J^j given in Eq. (57) is compound, with submatrices (except in the corners) for which i and k label rows and columns, respectively. For a three-dimensional orthogonal coordinate system, the matrices U, F, and J may be expanded as follows:

$$U = \begin{bmatrix} \rho \\ \rho u^1 \\ \rho u^2 \\ \rho u^3 \\ e \end{bmatrix} \qquad F^j = \begin{bmatrix} \rho u^j \\ \rho u^1 u^j + g^{1j} P \\ \rho u^2 u^j + g^{2j} P \\ \rho u^3 u^j + g^{3j} P \\ u^j (e + P) \end{bmatrix} \tag{61}$$

$$J^j = \begin{bmatrix} 0 & \delta^j_1 & \delta^j_2 & \delta^j_3 & 0 \\ -u^1u^j + g^{1j}\frac{\partial P}{\partial \rho} & u^j + (1 - \Gamma)\,\delta^j_1 u^1 & u^1\delta^j_2 - \Gamma g^{1j} g_{22} u^2 & u^1\delta^j_3 - \Gamma g^{1j} g_{33} u^3 & g^{1j}\Gamma \\ -u^2u^j + g^{2j}\frac{\partial P}{\partial \rho} & u^2\delta^j_1 - \Gamma g^{2j} g_{11} u^1 & u^j + (1 - \Gamma)\delta^j_2 u^2 & u^2\delta^j_3 - \Gamma g^{2j} g_{33} u^3 & g^{2j}\Gamma \\ -u^3u^j + g^{3j}\frac{\partial P}{\partial \rho} & u^3\delta^j_1 - \Gamma g^{3j} g_{11} u^1 & u^3\delta^j_2 - \Gamma g^{3j} g_{22} u^2 & u^j + (1 - \Gamma)\delta^j_3 u^3 & g^{3j}\Gamma \\ -\frac{1}{\rho}u^j(e + P) + u^j\frac{\partial P}{\partial \rho} & \frac{e + P}{\rho}\delta^j_1 - u^j\Gamma g_{11} u^1 & \frac{e + P}{\rho}\delta^j_2 - u^j\Gamma g_{22} u^2 & \frac{e + P}{\rho}\delta^j_3 - u^j\Gamma g_{33} u^3 & u^j(1 + \Gamma) \end{bmatrix} \tag{62}$$

The eigenvalues of J^j are u^j, u^j, u^j, $u^j + c^j$, and $u^j - c^j$, where $c^j = c\sqrt{g^{jj}}$. Then

$$\frac{\partial U}{\partial t} + J^j U_{,j} = 0 \tag{63}$$

represents a further alternative formulation for source-free, nonviscous systems. The linearized hydrodynamic equations result when J^j is taken to be constant.

The Lax-Wendroff pseudoviscosity matrix, developed for the one-dimensional case (Ref. 1), is a quadratic function of J constructed so that the eigencolumns of J corresponding to the eigenvalues u, u + c, and u - c are also eigencolumns of Q corresponding to the eigenvalues q_1, q_2, and q_3, respectively:

$$\begin{aligned} Q = q_1 &\left[\frac{J - (u + c)I}{-c}\right]\left[\frac{J - (u - c)I}{c}\right] \\ + q_2 &\left[\frac{J - uI}{c}\right]\left[\frac{J - (u - c)I}{2c}\right] \\ + q_3 &\left[\frac{J - uI}{-c}\right]\left[\frac{J - (u + c)I}{-2c}\right] \end{aligned} \tag{64}$$

where I is the unit matrix. Directional indices are understood for Q, q, J, u, and c, as in Eq. (64), but need not be shown in a one-dimensional formula. The eigenvalues q_1, q_2, and q_3 are chosen as positive-definite expressions

$$
\begin{aligned}
q_1 &= \kappa \left| \xi \frac{\partial u}{\partial x} \right| \\
q_2 &= \kappa \left| \xi \frac{\partial}{\partial x} (u + c) \right| \\
q_3 &= \kappa \left| \xi \frac{\partial}{\partial x} (u - c) \right|
\end{aligned}
\tag{65}
$$

where ξ is a coefficient with the dimensions of x (later to be identified with a basic mesh interval) and $\kappa \sim 1$ is an arbitrary numerical coefficient.

Equation (55) is then modified to read

$$
\frac{\partial U}{\partial t} + F_{,x} = \Phi_{,x} = \frac{\partial}{\partial x} \Phi \tag{66}
$$

where

$$
\Phi = 1/2 Q \xi U_{,x} \tag{67}
$$

is to be treated like a scalar under covariant differentiation. Equation (64) may also be written

$$
Q = a_0 I + a_1 J + a_2 J^2 \tag{68}
$$

where

$$
\begin{aligned}
a_0 &= [-q_1(u^2 - c^2) + 1/2 q_2 u(u - c) + 1/2 q_3 u(u + c)]/c^2 \\
a_1 &= [2q_1 u - 1/2 q_2(2u - c) - 1/2 q_3(2u + c)]/c^2 \\
a_2 &= [-q_1 + 1/2 q_2 + 1/2 q_3]/c^2
\end{aligned}
\tag{69}
$$

In order to avoid calculation of the Jacobian in second-order difference approximations to Eqs. (1), (2), and (3), it is customary to employ two-step time differencing (Ref. 2). In this case, a procedure attributed to Lapidus may be used to eliminate the Jacobian from the pseudoviscosity terms as well. The following derivation shows one way in which this result can be reached.

Equations (66), (67), and (68) yield

$$
\frac{\partial U}{\partial t} + F_{,x} = 1/2 \partial_x (a_0 I + a_1 J + a_2 J^2) \xi U_{,x} \tag{70}
$$

or

$$\frac{\partial U}{\partial t} + \overline{F}_{,x} = 1/2\partial_x (a_0 \xi U_{,x}) \tag{71}$$

where

$$\overline{F} = F - 1/2 (a_1 J \xi U_{,x} + a_2 J \xi F_{,x}) \tag{72}$$

A time-difference equation approximating Eq. (71) may be written

$$U^{n+1} = U^n - \Delta t \, \overline{F}^{n+\frac{1}{2}}_{,x} + 1/2\Delta t \frac{\partial}{\partial x} \xi (a_0 U_{,x})^{n+\frac{1}{2}} \tag{73}$$

The pseudoviscosity terms, being of order $\xi^2 \Delta t$, need not be centered. Thus, from Eq. (72),

$$\overline{F}^{n+\frac{1}{2}} - F^{n+\frac{1}{2}} = -1/2 J^n \xi (a_1 U_{,x} + a_2 F_{,x})^n \tag{74}$$

Defining a new quantity $U^{n+\frac{1}{2}}$ such that

$$\overline{F}^{n+\frac{1}{2}} - F^n = J^n (U^{n+\frac{1}{2}} - U^n) \tag{75}$$

and then subtracting Eq. (74) from Eq. (75) yields

$$F^{n+\frac{1}{2}} - F^n = J^n [U^{n+\frac{1}{2}} - U^n + 1/2\xi (a_1^n U^n_{,x} + a_2^n F^n_{,x})] \tag{76}$$

Here, $F^{n+\frac{1}{2}} = F(U_0^{n+\frac{1}{2}})$, where U_0 is the solution of Eq. (55), i. e., the undamped case. Therefore,

$$F^{n+\frac{1}{2}} - F^n = \frac{\Delta t}{2} J^n \frac{\partial U_0^n}{\partial t} = - \frac{\Delta t}{2} J^n F^n_{,x} \tag{77}$$

From Eqs. (76) and (77), the difference equation for $U^{n+\frac{1}{2}}$ is obtained:

$$U^{n+1/2} = U^n - 1/2\Delta t F^n_{,x} - 1/2\xi (a_1 U^n_{,x} + a_2 F^n_{,x}) \tag{78}$$

Equations (78) and (73), with the relation

$$\overline{F}^{n+\frac{1}{2}} = F(U^{n+\frac{1}{2}}) \tag{79}$$

form an explicit time-difference equation system with Lax-Wendroff pseudoviscosity, in the two-step form, for the one-dimensional case.

Unfortunately, since the J and Q matrices corresponding to different directions fail to commute, the generalization of the Lax-Wendroff pseudoviscosity to two or more space dimensions is not straightforward. By superposition of the above one-dimensional forms (Ref. 3), the system can be written (with sum convention suppressed) as

$$U^{n+\frac{1}{2}} = \left[U - 1/2\Delta t\sum_j \partial_j F^j - 1/2\sum_j \xi^j a_1^j U_{,j} - 1/2\sum_j \xi^j a_2^j F^j_{,j}\right]^n \tag{80}$$

$$U^{n+1} = U^n - \Delta t\sum_j \partial_j (F^j - 1/2 a_0^j \xi^j U_{,j})^{n+\frac{1}{2}} \tag{81}$$

In evaluating the covariant derivatives for the axially symmetric case, it should be noted that $a_0^{(2)} = a_1^{(2)} = a_2^{(2)} = 0$. Therefore, in this case the derivatives within the pseudoviscosity terms, denoted by commas, are simple derivatives containing no curvature terms. The expression $a_0^j \xi^j U$, like Φ^j, is not to be treated as a vector or tensor field when being differentiated. The a_p^j are defined by Eqs. (65) and (69), with a directional superscript j (not subject to the sum convention) applied to u, c, x, q, ξ, and perhaps κ.

A simplified formulation, attributed (Ref. 2) to Lapidus, is obtained by annulling a_1^j and a_2^j and setting a_0^j equal to q_1^j:

$$Q^j = a_0^j I = \kappa |\xi^j \frac{\partial}{\partial x^j} u^j| I \quad \text{(no sum convention)} \tag{82}$$

The viscous stress τ^{ij} can be accounted for in the definition of F as in Eqs. (40) and (48), or by introducing on the right side of Eqs. (55) and (63) a dissipative term $V^j_{,j}$, where

$$V^j = \begin{bmatrix} 0 \\ [\tau^{ij}] \\ g_{\alpha\beta} \dfrac{m^\beta}{\rho} \tau^{\alpha j} \end{bmatrix} \tag{83}$$

The Jacobian $J^j_\tau = \partial V^j / \partial U$ is

$$J^j_\tau = \begin{bmatrix} 0 & [0] & 0 \\ [\partial\tau^{ij}/\partial\rho] & [\partial\tau^{ij}/\partial m^k] & [\partial\tau^{ij}/\partial e] \\ g_{\alpha\beta}\dfrac{m^\beta}{\rho}\left(\dfrac{\partial\tau^{\alpha j}}{\partial\rho} - \dfrac{\tau^{\alpha j}}{\rho}\right) & \left[g_{\alpha\beta}\dfrac{m^\beta}{\rho}\dfrac{\partial\tau^{\alpha j}}{\partial m^k} + g_{\alpha k}\dfrac{\tau^{\alpha j}}{\rho}\right] & g_{\alpha\beta}\dfrac{m^\beta}{\rho}\partial\tau^{ij}/\partial e \end{bmatrix} \tag{84}$$

It is apparent that J^j_τ has at least one zero eigenvalue, due to the absence of τ^{ij} in Eq. (1). A pseudoviscosity of this form, based upon a suitable redefinition of τ^{ij}, is thus relatively ineffective for stabilizing Eulerian difference equations against nonlinear instabilities such as those investigated by Burstein (Ref. 3). However, the Richtmyer-von Neumann pseudoviscosity, which is of this form, has long been used for Lagrangian equations, and variants have been successfully used for Eulerian cases, such as the following:

$$\tau^{11} = 2\nu^{(1)} \frac{\partial u}{\partial r}, \quad r^2\tau^{22} = 2\nu^{(1)} \frac{u}{r}, \quad \tau^{33} = 2\nu^{(3)} \frac{\partial v}{\partial y}$$

where

$$\nu^{(1)} = \kappa\rho\xi^{(1)}\mathrm{Max}\left(0, -\xi^{(1)} \frac{\partial u}{\partial r}\right), \quad \nu^{(3)} = \kappa\rho\xi^{(3)}\mathrm{Max}\left(0, -\xi^{(3)} \frac{\partial v}{\partial y}\right)$$

TWO-STEP LAX-WENDROFF DIFFERENCE EQUATIONS WITH AXIAL SYMMETRY

In formulating the space-time difference equations, it is convenient to avoid half-integer indices by ascribing even integer indices to the basic mesh quantities and odd indices to the midpoints. The basic time difference

Δt introduced in the preceding section will therefore be replaced by $2\Delta t$ in this and subsequent sections, and the space differences ξ^1 and ξ^3 by

$$2\delta r_i = r_{i+1} - r_{i-1} \tag{85}$$

$$2\delta y_j = y_{j+1} - y_{j-1} \tag{86}$$

respectively. The following equations are written for even values of the space-time parity index i+j+n; however, it will be noted that both curvature and pseudoviscosity terms depend upon odd-parity values of U, such as $U^n_{i+1,j}$. The definitions of F, G, and H given in Eq. (48) permit elimination of tensor notation. For a uniform rectangular spatial mesh,

$$U^n_{i+1,j} = \frac{1}{4}\left(U^n_{i+2,j} + U^n_{ij} + U^n_{i+1,j+1} + U^n_{i+1,j-1}\right) \tag{87}$$

$$\begin{aligned} U^{n+1}_{i+1,j} = {} & U^n_{i+1,j} - 1/2\left[\frac{\Delta t}{\delta r_{i+1}} + (a_2^{(1)})^n_{i+1,j}\right](F^n_{i+2,j} - F^n_{ij}) - 1/2\left[\frac{\Delta t}{\delta y_j} + (a_2^{(3)})^n_{i+1,j}\right] \\ & (G^n_{i+1,j+1} - G^n_{i+1,j-1}) - 1/2\Delta t\, H^n_{i+1,j} - 1/2(a_1^{(1)})^n_{i+1,j}(U^n_{i+2,j} - U^n_{ij}) \\ & - 1/2(a_1^{(3)})^n_{i+1,j}(U^n_{i+1,j+1} - U^n_{i+1,j-1}) \end{aligned} \tag{88}$$

$$U^{n+1}_{ij} = 1/4\left(U^{n+1}_{i+1,j} + U^{n+1}_{i-1,j} + U^{n+1}_{i,j+1} + U^{n+1}_{i,j-1}\right) \tag{89}$$

$$\begin{aligned} U^{n+2}_{ij} = {} & U^n_{ij} - \frac{\Delta t}{\delta r_i}(F^{n+1}_{i+1,j} - F^{n+1}_{i-1,j}) - \frac{\Delta t}{\delta y_i}(G^{n+1}_{i,j+1} - G^{n+1}_{i,j-1}) - \Delta t\, H^{n+1}_{ij} \\ & + 1/2\frac{\Delta t}{\delta r_i}[(a_0^{(1)})^{n+1}_{i+1,j}(U^{n+1}_{i+2,j} - U^{n+1}_{ij}) - (a_0^{(1)})^{n+1}_{i-1,j}(U^{n+1}_{ij}) - U^{n+1}_{i-2,j})] \\ & + 1/2\frac{\Delta t}{\delta y_i}[(a_0^{(3)})^{n+1}_{i,j+1}(U^{n+1}_{i,j+2} - U^{n+1}_{ij}) - (a_0^{(3)})^{n+1}_{i,j-1}(U^{n+1}_{ij} - U^{n+1}_{i,j-2})] \end{aligned} \tag{90}$$

Equations (87) and (89) may be used to eliminate odd-parity quantities from Eqs. (88) and (90), or, alternatively, the four equations may be solved in sequence. In the latter case, both even- and odd-parity values of U must

be simultaneously available, but the amount of computing required is reduced substantially. A third possibility is to modify the equations so that differences of odd-parity quantities, for example, are replaced by differences of even-parity quantities. This last approach is more complex in some ways and will therefore not be followed in this report.

A related problem is generalization to a nonuniform rectangular mesh. Let

$$\begin{aligned} \Delta r_i &= r_{i+1} - r_i \qquad & \nabla r_i &= r_i - r_{i-1} = \Delta r_{i-1} \\ \Delta y_j &= y_{j+1} - y_j \qquad & \nabla y_j &= y_j - y_{j-1} = \Delta y_{j-1} \end{aligned} \tag{91}$$

Then Eqs. (87) and (89) become, respectively,

$$U^n_{i+1,j} = \left(\frac{1}{\Delta r_{i+1}} + \frac{1}{\nabla r_{i+1}} + \frac{1}{\Delta y_j} + \frac{1}{\nabla y_j}\right)^{-1} \left(\frac{1}{\Delta r_{i+1}} U^n_{i+2,j} + \frac{1}{\nabla r_{i+1}} U^n_{ij} + \frac{1}{\Delta y_j} U^n_{i+1,j+1} + \frac{1}{\nabla y_j} U^n_{i+1,j-1}\right) \tag{92}$$

and

$$U^{n+1}_{ij} = \left(\frac{1}{\Delta r_i} + \frac{1}{\nabla r_i} + \frac{1}{\Delta y_j} + \frac{1}{\nabla y_j}\right)^{-1} \left(\frac{1}{\Delta r_i} U^{n+1}_{i+1,j} + \frac{1}{\nabla r_i} U^{n+1}_{i-1,j} + \frac{1}{\Delta y_j} U^{n+1}_{i,j+1} + \frac{1}{\nabla y_j} U^{n+1}_{i,j-1}\right) \tag{93}$$

These equations, like Eqs. (87) and (89), are accurate to first order in the mesh interval.

In Eq. (88), the quantities H, $a_1^{(1)}$, $a_1^{(3)}$, $a_2^{(1)}$, and $a_2^{(3)}$ appear with odd parity. H may be most easily evaluated from the odd-parity U given by Eq. (92). From Eqs. (65) and (69), the a's consist of differences of even-parity values of u, v, and c, with odd-parity coefficients involving the same quantities. These coefficients may be evaluated as one-directional weighted means of the adjacent even-parity quantities:

$$(a_1^{(1)})_{ij}^{n+1} = \{[2q_1^{(1)}\overline{u} - 1/2q_2^{(1)}(2\overline{u} - \overline{c}) - 1/2q_3^{(1)}(2\overline{u} + \overline{c})]/\overline{c}^2\}_{ij}^{n+1} \quad (94)$$

etc., where the bar denotes the appropriate one-directional mean:

$$\overline{u}_{ij}^{n+1} = \left(\frac{1}{\Delta r_i} + \frac{1}{\nabla r_i}\right)^{-1} \left(\frac{1}{\Delta r_i} u_{i+1,j}^{n+1} + \frac{1}{\nabla r_i} u_{i-1,j}^{n+1}\right) \quad (95)$$

etc. In Eq. (90), the odd-parity H_{ij}^{n+1} may be computed from the U_{ij}^{n+1} given by Eq. (93), differences of which also appear as coefficients of even-parity a_0. These a_0 depend upon even-parity q's which may be defined by differencing $\overline{u}$, $\overline{v}$, and $\overline{c}$ as given by formulae like Eq. (95). For the viscous stress components, contributions to F and G are always of even parity and contributions to H of odd parity. The differencing, in this case, should always be based upon the U values with the opposite parity and the same time index.

BOUNDARY CONDITIONS FOR EXPLICIT DIFFERENCE EQUATIONS

On reflective boundaries (including always the symmetry axis), the normal component of velocity vanishes, while the tangential component of velocity and the "scalars" $a_0^{(j)}$, $a_1^{(i)}$, $a_2^{(j)}$, ρ, e, P, etc., all have vanishing normal derivatives. On the axis, the quantity u/r is equal to $\partial u/\partial r$. Therefore, the first, third, and fourth components of F and the second components of U, G, and H vanish on the axis.

Equations (88), (90), (92), and (93) explicitly refer to points which may lie outside the active mesh. Therefore, a single row of mesh points may be provided outside the active region in order to avoid rewriting these equations for boundary points. Values of quantities assigned to these external points are identical to those one mesh spacing within, except that for reflective boundaries the normal component of velocity is reversed in sign. Values of H are not required at the external points. Other quantities which vanish on the reflective boundary are antisymmetric with respect to the boundary, while the remaining quantities (whose normal derivatives

vanish on the boundary) are symmetric. Differentiation (ordinary or covariant) of a quantity in the normal direction reverses its symmetry character at a reflective boundary.

Boundaries other than the symmetry axis may be defined in various ways; a free boundary is one on which all quantities at external points vanish, whereas a symmetric transmittive boundary is one on which all quantities have vanishing normal derivatives, and an extrapolative boundary is one on which the normal second derivatives of all quantities vanish.

STABILITY OF LAX-WENDROFF DIFFERENCE EQUATIONS

The stability of the linearized Lax-Wendroff equations (Eq. (63) with constant J^j) is analyzed in reference 2, where a sufficient condition for stability is found to be

$$(|\underline{u}| + c)\frac{\Delta t}{\Delta x} < \frac{1}{\sqrt{2}} \tag{96}$$

for the case in which $\delta y = \delta r = \Delta x$. The analysis is more complex for the case of a nonuniform rectangular mesh, but the pair of conditions

$$(|u| + c)\frac{\Delta t}{\delta r} < \frac{1}{\sqrt{2}} \tag{97}$$

$$(|v| + c)\frac{\Delta t}{\delta y} < \frac{1}{\sqrt{2}} \tag{98}$$

can be taken as essentially equivalent. The effect of the viscosity terms is very complex, but Burstein has shown (Ref. 3) that the a_2 contributions probably increase stability slightly, whereas it is shown in Ref. 2 that the effect of a constant a_0 is slightly destabilizing. The simple viscosity given by Eq. (82) is always stabilizing, however, and the same is probably true of the complete formulation in the nonlinear case.

The coefficient κ which appears in the pseudoviscous terms can be modified if the calculation runs into difficulties such as negative internal

energy. In order for this to be possible, the contribution of the pseudo-viscous terms at each point must be temporarily saved while the internal energy is calculated. Since each component of U is linear in κ, it is possible to write

$$\rho = \rho_0 + \kappa \frac{\partial \rho}{\partial \kappa} \qquad n = n_0 + \kappa \frac{\partial n}{\partial \kappa}$$
$$m = m_0 + \kappa \frac{\partial m}{\partial \kappa} \qquad e = e_0 + \kappa \frac{\partial e}{\partial \kappa} \tag{99}$$

where the subscript zero denotes the value without pseudoviscosity. Then

$$\begin{aligned} 2\rho e - m^2 - n^2 &= 2\rho_0 e_0 - m_0^2 - n_0^2 + 2\kappa\left(\rho_0 \frac{\partial e}{\partial \kappa} + e_0 \frac{\partial \rho}{\partial \kappa} - m_0 \frac{\partial m}{\partial \kappa} - n_0 \frac{\partial n}{\partial \kappa}\right) \\ &\quad + \kappa^2 \left[2 \frac{\partial \rho}{\partial \kappa}\frac{\partial e}{\partial \kappa} - \left(\frac{\partial m}{\partial \kappa}\right)^2 - \left(\frac{\partial n}{\partial \kappa}\right)^2\right] \\ &= C + 2\kappa B + \kappa^2 A \end{aligned} \tag{100}$$

The value of κ required to bring the internal energy to zero is thus

$$\kappa = \frac{- B \pm \sqrt{B^2 - AC}}{A} \tag{101}$$

where, assuming that the pseudoviscosity has a genuine dissipative effect, the sign can be chosen so that κ is positive as well as real. A higher positive value of κ will result in a positive internal energy; a difference approximation to Eq. (21) could be used to estimate an appropriate value.

SUMMARY AND CONCLUSION

The basic equations of the method are Eqs. (88) and (90). The former is solved on odd-numbered time steps, the latter on even-numbered time steps. Values of U, i.e., the four conserved densities ρ, $\rho\mu$, ρv, and e, are computed for just half of the points in the spatial mesh in each time step, namely the points with indices i, j in time step n such that the "parity index" i + j + n is an even integer. Values of U for odd-parity points are

treated differently; where needed, they are computed from the even-parity values by interpolation, through the use of Eqs. (92) and (93).

The quantities F, G, and H appearing in Eqs. (88) and (90) are defined in Eq. (48). They depend upon the components of U; upon pressure P, which depends upon components of U (cf. Eqs. (6) and (4)); upon viscous stress τ, which depends upon differences of components of U (cf. Eq. (5)) and upon viscosities μ and μ_B, which may be functions of state and kinematic variables; and finally upon heat flux $\vec{q}$, which must be independently prescribed but may also depend upon state and kinematic variables.

The pseudoviscosity quantities a_o, a_1, and a_2 appearing in Eqs. (88) and (90) depend upon velocity components and their differences; upon the sound speed; and upon a parameter κ, which may be constant or under program control. The definitions are given in Eqs. (65) and (69). A simpler formulation is presented in Eq. (82). The purpose of these terms is to improve the stability characteristics of the difference equations; they have no physical significance.

The entire method outlined in this section is the subject of current research at many laboratories and is far from being a panacea for all problems which arise in the calculation of two-dimensional flow. However, development of this approach, as an alternative to the method employed in the OIL family of codes, is expected to improve the treatment of vapor-phase flow which can undergo very large density changes.

SECTION II

RADIATIVE TRANSFER IN HECTIC

INTRODUCTION

The immediate objective of the current work on radiative transfer in two-dimensional geometries has been to develop numerical methods that can be applied in HECTIC. The first step is to find a way to solve the equation

$$\vec{\Omega}\cdot\nabla I(\vec{r}, \vec{\Omega}) + \sigma(\vec{r})\, I(\vec{r}, \vec{\Omega}) = S(\vec{r}, \vec{\Omega}) \tag{102}$$

where I, S, and σ are, respectively, intensity, source, and absorption coefficients as functions of position r and direction Ω. Thus, at this step, polarization, frequency, and retardation are ignored. The source S is assumed known, so such problems as scattering are also deferred.

Descriptions of two codes that were partially developed for solving Eq. (102) as subroutines of HECTIC are given in reference 4. The first, TDRAD, uses the nonequilibrium diffusion method and the second, LONG2, uses the method of long characteristics. The current work, in general terms, has been devoted to:

1. Extending and refining the long-characteristics code by allowing a more flexible scheme for distributing rays and by providing a means of treating regions in which the diffusion approximation is valid.
2. Surveying methods for solving the elliptic difference equations that arise in two-dimensional versions of the nonequilibrium diffusion method.
3. Developing a new method, called the "view-factor" method, which avoids the most serious difficulties encountered in the other two.

The present report contains detailed descriptions of LONG2, the code that uses the extended method of characteristics, and SHORT2, the code that implements the view-factor method. Work performed on TDRAD is also reported. Specifically, descriptions of three algorithms for solving the nonequilibrium diffusion equations and the results of the tests to which they were put are presented.

An attempt has been made to keep the report reasonably self-contained, even at the expense of repeating much information that can be found elsewhere.

DEFINITIONS AND NOTATION

It is appropriate to establish at the outset various notations and conventions used in the discussion of the various routines for calculating radiative transfer in HECTIC. The physical systems these routines are designed to handle must be axially symmetric. Let the axis of symmetry be the z-axis of an x-y-z cartesian coordinate system. Let $r = (x^2 + y^2)^{\frac{1}{2}}$ denote the distance from the axis of the point with coordinates (x, y, z). The z-axis is called "vertical" and grad z is directed upward. Planes z = constant are horizontal. As might be expected, grad r points outward and its negative, inward. The region of interest lies inside a cylinder r = R and between two planes $z = z_{min}$ and $z = z_{max}$. This region is divided into a number of subregions, called "zones", by a set of cylinders $r = r_i$, where $1 \leq i \leq i_{max}$, and a set of planes $z = z_j$, where $0 \leq j \leq j_{max}$. As usual, $r_{i-1} < r_i$ and $z_{j-1} < z_j$.

Thus, a zone is a toroid of rectangular cross section. The zones fill the region so $r_{i_{max}} = R$, $z_0 = z_{min}$, and $z_{j_{max}} = z_{max}$. In the current FORTRAN code for HECTIC, the zones are numbered from 2 to IMAX*JMAX+1, where IMAX = i_{max} and JMAX = j_{max}. But in this part of the report, a double subscript is used and $Z_{i,j}$ denotes the zone bounded by

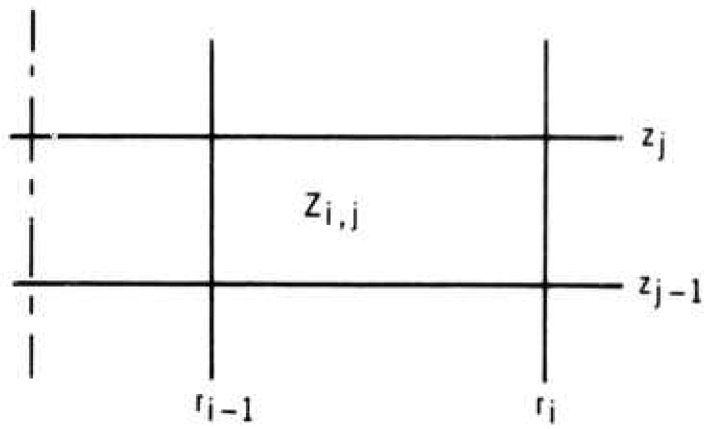


Figure 1. A Typical Zone in HECTIC

the cylinders $r = r_{i-1}$ and $r = r_i$ and the planes $z = z_{j-1}$ and $z = z_j$. (See Fig. 1.) The surface of zone $Z_{i,j}$ has four connected pieces, called "boundaries":

$BL_{i,j}$, lying on the cylinder $r = r_{i-1}$

$BR_{i,j}$, lying on the cylinder $r = r_i$

$BB_{i,j}$, lying on the plane $z = z_{j-1}$

$BA_{i,j}$, lying on the plane $z = z_j$

Let $\hat{z}$ = grad z, a unit vector parallel to the axis in the system, let $\hat{r}$ = grad r, a unit vector orthogonal to $\hat{z}$ and directed outward along a line through the axis, and let $\hat{z} \times \hat{r}$ be the third unit vector of the local basis.

If $\vec{Q}$ is the vector connecting the origin of coordinates to the point in question, then

$$\vec{P} = \vec{Q} + s\vec{\Omega} \tag{103}$$

is the parametric equation of a ray through the point. The vector $\vec{\Omega}$ is the direction of the ray. Directions are required to be unit vectors, and consequently the parameter s in Eq. (103) is the distance from Q to P(s). With respect to the local basis about the point P(s),

$$\vec{\Omega} = \mu\hat{z} - (1 - \mu^2) \quad (\hat{r} \cos \phi + \hat{r} \times \hat{z} \sin \phi) \tag{104}$$

where

$$\mu = \vec{\Omega} \cdot \hat{z}$$
$$\phi = \arctan (\vec{\Omega} \cdot \hat{r} \times \hat{z} / \vec{\Omega} \cdot r) \tag{105}$$

μ is called the "axial projection" of the ray and ϕ is called its "azimuth" at the point P. Since the direction of the local basis vector $\hat{r}$ depends on the coordinates of P, the magnitude of the azimuth ϕ depends on s, as may be seen from figure 2. When s is negative and large in magnitude, the horizontal component of $\vec{\Omega}$ points almost directly inward and ϕ has a small positive value. As s increases, so does ϕ. When $\phi < \pi/2$, the point is approaching the axis, and when $\phi > \pi/2$, it is receding from the axis. On the other hand, the axial projection μ of the ray depends only on its direction $\vec{\Omega}$ and does not vary with s.

The range $-\pi \leq \phi \leq 0$ is occupied by the azimuths of rays that pass the axis moving clockwise as viewed from above. In view of the assumed axial symmetry of the physical system, the variation of radiant intensity along such a ray is the same as along its reflection in the plane y = 0 (or any other plane containing the axis for that matter). Consequently, such rays need not be considered.

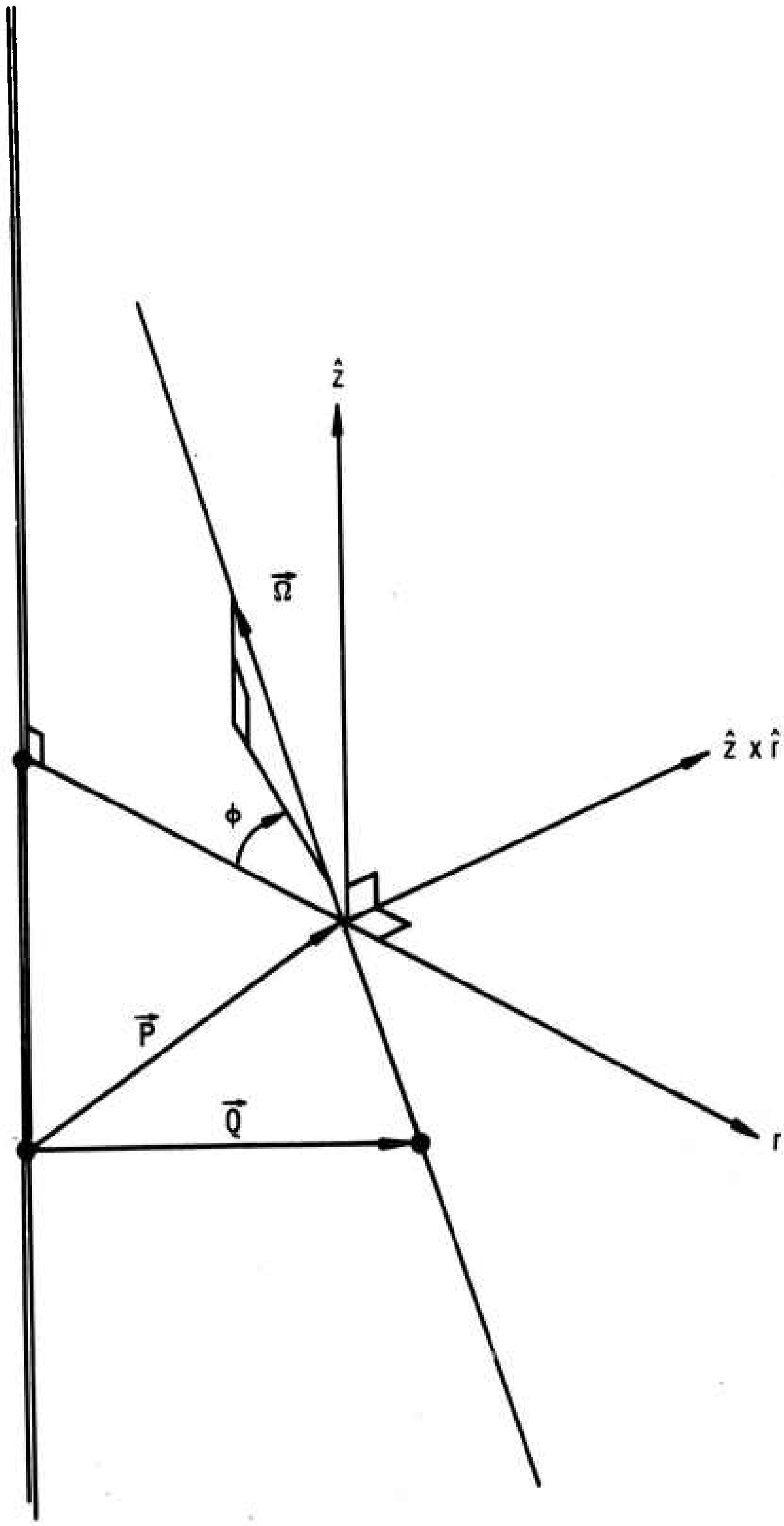


Figure 2. The Local Basis Vectors

SECTION III

THE LONG-CHARACTERISTIC METHOD

The idea behind the long-characteristic method is to calculate radiant intensity by means of the formula

$$I(\vec{b} + s\vec{\Omega}, \vec{\Omega}) = I(\vec{b}, \vec{\Omega})\, e^{-\int_0^s \sigma(\vec{b} + u\vec{\Omega})du} + \int_0^s S(\vec{b} + u\vec{\Omega}, \vec{\Omega})\, e^{-\int_u^s \sigma(\vec{b}+v\vec{\Omega})dv}\, du \tag{106}$$

where $I(\vec{r}, \vec{\Omega})$ is the intensity of radiation at $\vec{r}$ in direction $\vec{\Omega}$, $\sigma(\vec{r})$ is the absorption coefficient at $\vec{r}$, and $S(\vec{r}, \vec{\Omega})$ is the strength of the source at $\vec{r}$ in direction $\vec{\Omega}$. The specification of the intensity $I(\vec{b}, \vec{\Omega})$ at a point $\vec{b}$ on the boundary of the system in a direction $\vec{\Omega}$ that points into the system is a boundary condition. If, as is done in LONG2, the absorption coefficients and source strengths are taken to be constant within the hydro zones of HECTIC, the integrals of Eq. (106) are elementary. The problem that then remains is to determine the rate of energy deposition from a given distribution of intensities. The solution adopted in the subroutines of LONG2 is described below.

A SIMPLIFIED VERSION

Before the details of the long-characteristic method as implemented in LONG2 are discussed, a simplified version will be presented. The first step is to select a set of rays that are distributed through the physical

system more or less uniformly both in space and direction. In the simple version, this is done as follows. Let N be a large integer and let d be a length that is small compared with the dimensions of the zones in the HECTIC hydro mesh. Let

$$\mu_n = \frac{N-n+\frac{1}{2}}{N}, \qquad \eta_n = (1-\mu_n^2)^{\frac{1}{2}} \tag{107}$$

for $1 \leq n \leq 2N$. Then to each triplet of integers, ℓ, m, n for which $1 \leq n \leq N$, a ray $R_{\ell,m,n}$ is associated whose parametric equations are

$$\begin{aligned} x &= \eta_n s \\ y &= (\ell - \tfrac{1}{2})\, d \\ z &= \frac{m\, d}{\eta_n} + \mu_n s \end{aligned} \tag{108}$$

the parameter s being arc length. The restriction $\ell \leq R/d$ on ℓ guarantees that the ray comes close enough to the axis to pierce the cylinder $r = R$. There is some complicated restriction on m that will ensure that the ray actually does penetrate the region of interest, but it serves no useful purpose to write it down. The set of rays corresponding to a fixed value of n is called a "grid." The rays of a grid are all parallel and intersect any plane to which they are normal in a square lattice of points with a separation of d between nearest neighbors. The set of rays corresponding to a fixed value of n and ℓ forms a comb.

To verify that the complete set of rays is uniform, their distribution over some small element of volume will be analyzed. In view of the axial symmetry of the system, the element of volume should be a figure of revolution. Let it be a toroid of rectangular cross section: $r_o \leq r \leq r_o + \Delta r$, $z_o \leq z \leq z_o + \Delta z$, where $\Delta z << \Delta z$, $d << \Delta z$. Consider those rays whose directions have axial projections lying in the range $\mu_o \leq \mu \leq \mu_o + \Delta\mu$ that

intersect the volume element when their azimuths lie in the range $\phi_0 \le \phi \le \phi_0 + \Delta\phi$, and calculate the sum of the lengths of these intersections. The length of one such intersection is

$$t = \frac{\Delta r}{(1 - \mu_o^2)^{\frac{1}{2}} \cos \phi_o} \tag{109}$$

It can be seen from figure 3 that $r \sin \phi = y$ and hence that $r \cos\phi \, \Delta\phi = \Delta y$. Therefore, $\Delta \ell = \Delta y/d = r_o \cos \phi_o \, \Delta\phi/d$ is the number of ℓ values of rays that have their azimuths in the desired range at $r = r_o$. The vertical separation between $R_{\ell,m,n}$ and ray $R_{\ell,m+1,n}$ is d/η_n. Therefore, the number of m values of rays with fixed ℓ and n striking the volume element is $\Delta m = \Delta z/(d/\eta_n) = \eta_n \, \Delta\ell/d$. Finally, the number of n-values with μ in the desired range is simply $\Delta n = N\Delta\mu$. Thus, the sum T of the lengths of the intersections is

$$T = t \, \Delta\ell \, \Delta m \, \Delta n = \Delta r \, r_o \, \Delta\phi \, \Delta z \, N \, \Delta\mu/d^2$$

$$= \frac{N \, \Delta V \, \Delta\Omega}{2\pi d^2} \tag{110}$$

where $\Delta V = 2\pi \, r_o \, \Delta r \, \Delta z$ is the magnitude of the volume element and $\Delta\Omega = \Delta\phi \, \Delta\mu$ is the solid angle subtended by the range of directions being considered. Since the sum of the lengths is proportional to $\Delta V \, \Delta\Omega$ and independent of r_o, z_o, μ_o, and ϕ_o, one may conclude that the distribution is uniform in both space and direction.

The rate of deposition of radiant energy in a HECTIC hydro zone is calculated as the difference between the inward-going flux and the outward-going flux through the boundaries of the zone. The net inward flux through a vertical boundary of radius r and altitude ranging from z_1 to z_2 is

$$F = 2 \int_0^{\pi} \int_{-1}^{1} \int_{z_1}^{z_2} \eta \cos \phi \, I(r, z, \vec{\Omega}) \, 2\pi r \, dz \, d\mu \, d\phi \tag{111}$$

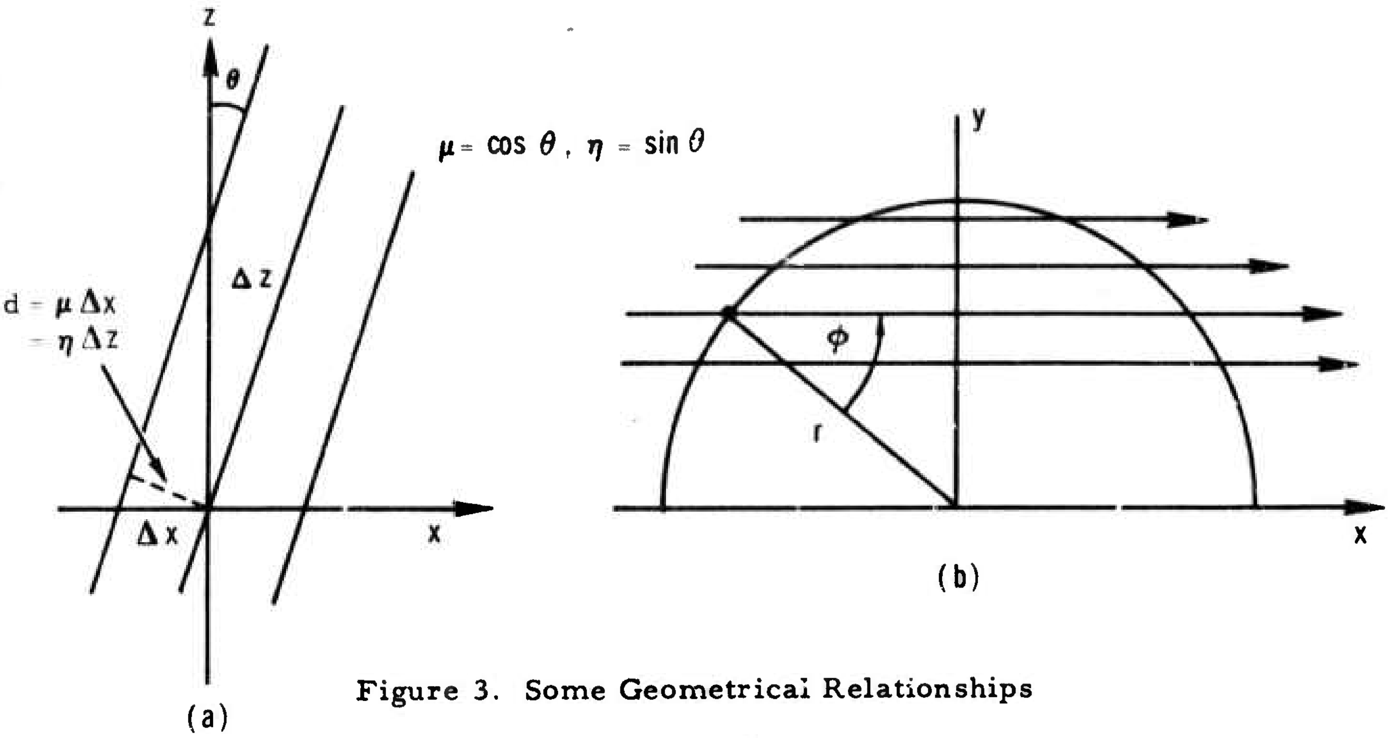


Figure 3. Some Geometrical Relationships

where $\eta = (1 - \mu^2)^{\frac{1}{2}}$. The value of the integral is estimated by the simplest method, i.e., by replacing the integral with finite sum:

$$F = 4\pi r \sum \eta \cos\phi \; I \, \Delta z \, \Delta\mu \, \Delta\phi \tag{112}$$

where the sum extends over all rays striking the surface. The vertical distance between rays is d/η, so it is appropriate to set $\eta \, \Delta z = d$ (see Fig. 3a). Next, referring to Eq. (108) and figure 3b, it is seen that $\sin\phi = y/r$, so $|\cos\phi| \, \Delta\phi = \Delta y/r = d/r$. Finally, $\Delta\mu = 1/N$. Thus, Eq. (112) becomes

$$F = \frac{4\pi d^2}{N} \sum \pm I \tag{113}$$

where the sign to use is that of $\cos\phi$: + for inward-going and - for outward-going rays. The net upward flux through a horizontal zone boundary with inner radius r_1, outer radius r_2, and altitude z is

$$F = 2 \int_0^{\pi} \int_{-1}^{1} \int_{r_1}^{r_2} \mu I(r, z, \vec{\Omega})\, 2\pi r\, dr\, d\mu\, d\phi \qquad (114)$$

Referring again to figure 3b, it can be seen

$$x = -r \cos \phi$$

$$y = r \sin \phi$$

and hence

$$\frac{\partial(r,\phi)}{\partial(x,y)} = \begin{vmatrix} \frac{\partial r}{\partial x} & \frac{\partial r}{\partial y} \\ \frac{\partial \phi}{\partial x} & \frac{\partial \phi}{\partial y} \end{vmatrix}$$

$$= \begin{vmatrix} \frac{x}{r} & \frac{y}{r} \\ \frac{y}{r^2} & -\frac{x}{r^2} \end{vmatrix}$$

$$= -\frac{x^2}{r^3} - \frac{y^2}{r^3} = -\frac{1}{r}$$

Therefore, $r\, dr\, d\phi = dx\, dy$. But, as before, $\Delta y = d$ and $|\mu| \Delta x = d$; so making the same sort of estimate of the flux F in Eq. (114) again yields Eq. (113), where this time the sign of I is that of μ : + for upward and - for downward. The rate of deposition of radiant energy in a HECTIC hydro zone is therefore

$$\dot{E} = \frac{4\pi d^2}{N} \sum (I_{in} - I_{out}) \qquad (115)$$

the sum extending over all rays that strike the zone.

THE IMPLEMENTED VERSION

The long-characteristic method, as implemented in LONG2, is a refinement of the simple version described above. Some of the principal changes are (1) more flexibility in the selection of rays, (2) the inclusion of some correction factors, and (3) a special treatment for optically thick zones.

One of the practical difficulties of the long-characteristic method is the large number of arithmetic operations it involves. It is necessary to calculate the change of intensity along every line segment resulting from the intersection of a ray with a hydro zone. It is therefore desirable to allow the widest possible latitude in the distribution of rays so that certain regions may be sampled more thoroughly than others.

In the simple version, a grid consists of rays that are parallel and intersect any orthogonal plane in a square lattice of points. The first generalization incorporated into LONG2 is a facility for having a higher density of rays in some regions than in others. As before, let 2N be the number of grids and let n be the index of a grid. For each n, $1 < n \le 2N$, there are selected two increasing sequences $(y_{\ell,n})_{\ell=1}^{L_n}$ and $(z^*_{m,n})_{m=1}^{M_n}$. With this n and every pair ℓ, m of integers is associated a ray $R_{\ell,m,n}$ whose parametric equations are

$$
\begin{aligned}
x &= \eta_n s \\
y &= 1/2(y_{\ell-1,n} + y_{\ell,n}) \qquad (116) \\
z &= 1/2(z^*_{m-1,n} + z^*_{m,n}) + \mu_n s
\end{aligned}
$$

which generalize Eq. (108). The convention $y_{o,n} = 0$ is built in. To visualize this arrangement of rays, consider the half-plane $x = 0$, $y \ge 0$ partitioned into rectangles by a series of horizontal lines $z = z^*_{m,n}$ and a series of vertical lines $y = y_{\ell,n}$. Grid n will then consist of a set of rays parallel to each other and to the plane $y = 0$. They will have axial projection μ_n, and each ray of the grid will pass through the center of one of the rectangles.

If $n > N$, grid n consists of the rays of grid $n - N$ traced in the reverse direction.

The μ_n's for $1 < n \leq N$ must be distinct and fall within the open interval $0 < \mu < 1$, but otherwise they are arbitrary. No generality is lost by requiring that they be in monotonic sequence, so assume that

$$1 = \mu_1 > \mu_2 > \mu_3 \dots > \mu_n > 0$$

$$\mu_{N+n} = -\mu_n \quad , \qquad 1 \leq n \leq N \tag{117}$$

In case $n = 1$, Eq. (116) is replaced by

$$\begin{aligned} x &= 0 \\ y &= 1/2(r_{i-1} + r_i) \\ z &= s \end{aligned} \tag{118}$$

so there are just i_{max} rays in this grid.

Equation (115) must be modified to handle the generalized distribution of rays properly. The factor d^2 occurring in that equation is the area of the unit square in the lattice of rays having a fixed μ_n. This factor must be replaced by

$$a_{\ell,m,n} = (y_{\ell,n} - y_{\ell-1,n})(z^*_{m,n} - z^*_{m-1,n})\eta_n \tag{119}$$

when $1 < n < 2N$, which is the area of rectangle associated with ray $R_{\ell,m,n}$, and

$$a_{i,n} = \pi(r_i^2 - r_{i-1}^2)/2 \tag{120}$$

when $n = 1$ or $2N$, which is the area of that half of radial interval i lying in the half-space $y \geq 0$. The factor $1/N$ in Eq. (115) in just $\Delta\mu_n$. Thus, the equation for the rate of deposition of radiant energy in any given hydro zone is estimated to be

$$\dot{E} = 4\pi \sum \Delta\mu_n \, a_{\ell,m,n} (I_{in} - I_{out})_{\ell,m,n} \tag{121}$$

where the sum extends over all rays that intersect the given zone and $(I_{in} - I_{out})_{\ell,m,n}$ denotes the difference between the incoming and outgoing intensities along ray $R_{\ell,m,n}$.

In the current version of LONG2, the input routine allows only one value of Δy_n and one value of Δz_n^* to be specified. Then, the parametric equations for ray $R_{\ell,m,n}$ are

$$
\begin{aligned}
x &= \eta_n s \\
y &= (\ell - 1)\Delta y_n \\
z &= z_{min} + (m + \tfrac{1}{2})\Delta z_n^* + \mu_n s
\end{aligned}
\qquad (121a)
$$

and

$$
a_{\ell,m,n} = \begin{cases} \eta_n \Delta y_n \Delta z_n^* & \ell > 1 \\ \eta_n \Delta y_n \Delta z_n^*/2 & \ell = 1 \end{cases} \qquad (121b)
$$

No connection between the μ_n's and the $\Delta\mu_n$'s has yet been established. What has been tacitly assumed is that the range $1 \geq \mu \geq -1$ is divided into abutting intervals and that the nth interval is of length $\Delta\mu_n$ and contains μ_n in its interior. Thus, if μ_n^* denotes the left endpoint of interval n, then

$$
\begin{aligned}
\mu_n^* &= \mu_{n-1}^* - \Delta\mu_n \\
\mu_o^* &= 1 \\
\mu_N^* &= 0 \\
\mu_{N+n}^* &= \mu_{N-n} \\
\mu_{2N}^* &= -1 \\
\mu_n^* \leq \mu_n &\leq \mu_{n-1}^*, \qquad 1 \leq n \leq 2N
\end{aligned}
\qquad (122)
$$

and

$$\sum_{n=1}^{N} \Delta\mu_n = 1$$

An obvious source of error connected with using Eq. (119) for estimating rates of energy deposition is the lack of any close connection between the set of rays passing through the system of interest and the HECTIC hydro mesh. If the set of rays corresponding to a particular n were sufficiently dense, then the volume V of any given zone would be nearly $2\Sigma_{\ell,m} t_{\ell,m,n} a_{\ell,m,n}$, where $t_{\ell,m,n}$ is the length of the intersection of ray $R_{\ell,m,n}$ and the given zone. The proof is simple. The half-space $y \geq 0$ may be thought of as filled with non-overlapping tubes of rectangular cross section, in each of which is a ray and $t_{\ell,m,n}\ a_{\ell,m,n}$ is very nearly the volume of the intersection of the tube containing ray $R_{\ell,m,n}$ and the given zone, providing that $a_{\ell,m,n}$ is small enough. Consequently,

$$\begin{aligned} V^* &= \sum_{\ell,m,n} t_{\ell,m,n}\ a_{\ell,m,n}\ \Delta\mu_n \\ &\approx \sum_n \tfrac{1}{2} V\ \Delta\mu_n \qquad (123) \\ &= V \end{aligned}$$

The ratio V/V*, which is denoted by A, is calculated for each hydro zone. Zones for which A < 1 intersect more than their share of rays, whereas other zones intersect less than their share. To compensate to some extent for the fact that $A \neq 1$, the strength of the radiative source in each zone is multiplied by A. As was mentioned earlier, the source strength is assumed to be constant within a hydro zone. Therefore, Eq. (102) becomes

$$I_{out} = I_{in}\, e^{-\sigma t} + AS \int_0^t e^{-\sigma(t-u)}\, du$$

which, when integrated, gives

$$I_{out} = I_{in} e^{-\sigma t} + \frac{AS}{\sigma}(1 - e^{-\sigma t}) \tag{124}$$

and

$$I_{in} - I_{out} = (I_{in} - \frac{AS}{\sigma})(1 - e^{-\sigma t})$$

for the contribution of one summand in Eq. (121). Consider what happens when $\sigma t \ll 1$ and $I_{in} = 0$. Then $1 - e^{-\sigma t} \rightarrow \sigma t$ and

$$\begin{aligned} \dot{E} &= 4\pi \sum \Delta\mu_n \, a_{\ell,m,n} \; - \frac{AS}{\sigma} \; \sigma t_{\ell,m,n} \\ &= -4\pi \, ASV^* \\ &= -4\pi \, SV \end{aligned}$$

exactly. Thus, including the factor A in Eq. (124) ensures that the correct amount of radiation is emitted in each zone.

The quantity

$$w_{\ell,m,n} = \Delta\mu_n \; a_{\ell,m,n} \tag{125}$$

is called the "weight" of the ray and

$$J = 4\pi wI \tag{126}$$

is (loosely) called the "power" of a ray of weight w at a point where its intensity is I. Multiplying Eq. (124) by $4\pi w$ yields

$$J_{out} = J_{in} e^{-\sigma t} + \frac{4\pi w AS}{\sigma}(1 - e^{-\sigma t}) \tag{127}$$

and Eq. (121) becomes simply

$$\dot{E} = \sum (J_{in} - J_{out}) \tag{128}$$

The performance of the simplified version of the method is poor when the thickness of the hydro zones is greater than a few mean free paths, because the assumption of a constant source strength within each zone implies a discontinuity at the zone boundaries which, in turn, causes the estimated rate of transfer of energy to be much too high. In LONG2, there is a special treatment of the transfer of energy between thick zones, a thick zone being one for which $\sigma(r_i - r_{i-1})$ and $\sigma(z_j - z_{j-1})$ are both larger than some input criterion C_λ, usually 5 mfp. In the event that two thick zones share a common boundary, all rays crossing that boundary are ignored and the rate of energy transfer between those two zones is estimated to be

$$\dot{E}_d = -\frac{4\pi}{3} A_b \frac{\Delta S/\sigma}{\Delta\tau} \qquad (129)$$

where A_b is the area of the common boundary, $\Delta(S/\sigma)$ is the difference in zone centered values of S/σ, $\Delta\tau$ is the optical distance between the zone centers, and the minus sign indicates that the direction of flow is counter to the source gradient.

Consider a thick-thin boundary, i.e., the interface between two zone, one of which is thick and the other is not. First, it may be stated that the treatment of a thin zone is unaffected by the thickness or thinness of its neighbors, Eq. (121) being used in all such cases. However, where a ray passes from a thin zone into a thick one, its power at that point is counted as a contribution to $\dot{E}$ for the thick zone. Where a ray emerges from a thick zone into a thin one, its power is set to

$$J = 4\pi w \, AS/\sigma \qquad (130)$$

where AS/σ is evaluated at the center of the thick zone and $-J$ is added to $\dot{E}$ for the thick zone.

SOURCES

Two types of sources are treated by LONG2: distributed sources and boundary sources. The distributed sources are derived from temperatures calculated in the hydro subroutines by the formula

$$S = \frac{ac}{4\pi} \theta^4 \sigma \left[P\left(\frac{h\nu_2}{\theta}\right) - P\left(\frac{h\nu_1}{\theta}\right) \right] \tag{131}$$

where ac/4 is Stefan's constant, θ is the temperature, σ is the absorption coefficient, $\nu_1 \leq \nu \leq \nu_2$ is the band of frequencies being considered, and

$$P(x) = \frac{15}{\pi^4} \int_o^x \frac{u^3 du}{e^u - 1} \tag{132}$$

is the integral of Planck's radiation function. As has already been mentioned, the absorption coefficient and the source are calculated for each zone using zone-central data and are assumed to be constant over the zone.

In the current version of LONG2, boundary temperatures are input on cards and remain constant from time step to time step; but they could just as well be calculated. In any case, the surface temperature is specified for each axial interval $z_{j-1} \leq z \leq z_j$ on the outer cylindrical surface $r = R$ of the system and for each radial interval $r_{i-1} \leq r \leq r_i$ on the bottom $z = z_{min}$ and top $z = z_{max}$. The rate at which energy enters the system through each of these intervals is taken to be

$$\dot{E}_{in} = \frac{ac}{4} \theta_{out}^4 \alpha \left[P\left(\frac{h\nu_2}{\theta_{out}}\right) - P\left(\frac{h\nu_1}{\theta_{out}}\right) \right] \tag{133}$$

where α is the area of the interval. The power carried into the system by a ray originating in some interval is assumed to be

$$J = \frac{w \dot{E}_{in}}{\sum w} \tag{134}$$

where $\sum w$ is the sum of the weights of all rays originating in that interval.

GENERAL DESCRIPTION OF LONG2

The program LONG2 consists of two major subroutines of HECTIC whose entry points are DRAW and TRAN2. DRAW is entered only when a new hydro mesh has been generated. The result of calling DRAW is to produce for each ray the sequence $(t_p, k_p)_{p=1}^{p_{max}}$, where p_{max} is the number of segments into which the ray is divided by boundaries of the hydro mesh, t_p is the length of segment p, and k_p is the index of the zone in which segment p lies. The order of the sequence is the order in which the segments are traversed as the ray is traced. The reverse of the sequence for one ray is the sequence of another ray running in the opposite direction along the same line. DRAW stores these sequences in an I/O file in blocks of up to 1023 words. There is a three-word prologue to each sequence specifying the weight w of the ray and the boundary intervals in which it begins and ends. When the physical system under consideration has a symmetry plane, it is necessary to consider what happens on only one side of it. In such cases, either $z = z_o$ or $z = z_{jmax}$ may be the symmetry plane -- but assume it is the former. Then DRAW sets $z_o^* = z_o$ so that the system of rays will have the same symmetry plane. Rays hitting the plane $z = z_o$ do not then emerge from the system but are reflected there. In symmetrical problems, rays for which $\mu = \pm 1$ require special treatment because they are their own images in a symmetry plane; TRAN2 is instructed not to retrace such rays.

As the ray sequences are generated, the terms in the summation for V* are accumulated so that when the DRAW file is complete, an array containing $4\pi A$ for each zone may be established. (See Eq. (123).) In a similar manner, the weights of rays intersecting various outer boundary intervals are accumulated for use in evaluating Eq. (134).

When subroutine TRAN2 is called, it begins by calculating absorption coefficients and sources for each hydro zone and the input power for each interval of the outer boundary of the system. It then runs through the

file of ray sequences, calculating the changes in the power level and contributions to zonal rates of deposition of energy. This is done by first using Eq. (134) to get the initial power in the ray on the basis of information contained in the prologue to the sequence to identify the starting interval. Then for each pair (t,k) in sequence, Eq. (127) is applied to estimate the change in J caused by traveling a distance t in zone k. The order of the sequence is the order in which the segments of path are encountered as the ray passes through the system, so it is unnecessary to store the J's. When J is calculated at some zone boundary, it suffices to subtract it from the running total for $\dot{E}$ for the zone the ray leaves and to add it to that of the zone it enters. Equation (127) is then used to find the change in J as the latter zone is traversed. When a thick zone is encountered, the current J is added to the $\dot{E}$ of that zone in the normal manner. But then, rather than update J, all segments are skipped until a thin zone is encountered, at which point Eq. (130) is used to estimate J and the normal is resumed. After all the sequences have been processed, the zone boundaries are systematically scanned for thick-thick interfaces which, when found, are assumed to transmit energy at a rate specified by Eq. (129). When all the transfer rates have been estimated, the changes in the internal energy of each zone are calculated along with the energy transfer across the outer boundary of the system. The subroutine concludes with the calculation of various summary data used for checking energy balance.

RELATIONSHIP OF LONG2 TO THE PREVIOUS VERSION

As has been stated above, the current version of LONG2 consists of two subroutines: DRAW and TRAN2. These two subroutines are very similar to those with the same names described in reference 4. There are only two significant differences between the old and new versions of LONG2. The first is the introduction into the current version of a facility for assigning weights to rays which makes it possible to use nonuniform

grids. The second difference is the special diffusion-theoretic treatment of thick-thick interfaces in the current version. Such basic features as the over-all organization, the arrangement of data on I/O files, and the flow of control remain virtually unaltered.

AVENUES FOR FURTHER DEVELOPMENT

In this subsection, the question of what might be done with LONG2 to enable it to perform more efficiently the job for which it is designed is examined. The discussion of possibilities for extending the code to include treatments of scattering, radiation pressure, and retardation is beyond the scope of this report. There are two problems that should receive high priority in any plans to continue the development of LONG2. The first concerns the fact that because the rays travel all the way through the system, it is rarely possible to space them closely enough in regions where close spacing is required without having many more rays than necessary everywhere else, particularly at large radii. As a consequence, an excessive amount of computing effort is spent on relatively uninteresting parts of the geometry. The second major problem arises from the fact that the source of radiation is assumed to be constant over a HECTIC hydro zone, which results in an unrealistically high value for the computed rate of transfer of radiant energy. The so-called "view factor method" discussed in later sections of the present report was designed specifically to overcome these obstacles. The remainder of this subsection is a discussion of ways in which LONG2 could be extended to handle them.

The uniformity with which the rays are distributed through the system is a matter of primary importance in the long-characteristic method. If a HECTIC hydro zone is intersected by fewer than its fair share of rays, it will tend to cool faster than it should because it will fail to receive its full share of the radiant energy emitted by its neighbors. The result of

running a problem with an inadequate supply of rays is thus an uneveness in the temperature distribution that persists through all time steps. On the other hand, if the system is too generously infused with rays, the file of ray data will usually be enormous and the time required for processing it prohibitive. Suppose, for example, that the system is divided into a 40-by-40 HECTIC mesh and that the center of each zone coincides with the center of a ray in each of three grids. In such a mesh, a typical ray would be divided into about 80 segments. At two words per segment, the length of the file ray data would be, roughly, $40 \times 40 \times 80 \times 3 \times 2 = 7.68 \times 10^5$ words, which would take about a minute to read on a fast tape drive. More significant is the fact that $3 \times 80 = 240$ segments/zone is about 20 times more than are needed on the average, even though the inner zones would be penetrated by only five or six rays. The situation is usually even worse for large problems and for problems with thin inner zones, because in such cases the number of segments per ray tends to be higher.

The most direct method for allowing the density of rays to be adjusted more or less arbitrarily and independently in various parts of the system is to provide a means for handling rays that have endpoints in the interior of the system. This could be accomplished by allowing a ray to split into two or more parallel subordinate rays. In TRAN2, when a ray being traced arrived at a point where it was split, the energy that it carried could be divided equally among the new subordinate rays to provide them with a sort of initial condition. In the reverse direction, the energy in the subordinate rays could be accumulated at the point of splitting to provide an initial condition for the reverse trace of the main ray. Such a modification, while simple in principle, would considerably increase the complexity of the subroutine DRAW. However, it seems likely that if such a splitting scheme were instituted, it would be possible to allow subordinate rays to be split again without additional complication. The subroutine TRAN2 would not necessarily be much affected by splitting, since the solutions of all the logical problems solved in DRAW could be transmitted to TRAN2 in the file of ray data.

The second of the above-mentioned problems is associated with the assumption of a flat source over each HECTIC zone. A partial solution has already been implemented in the current version of LONG2, namely, the use of diffusion theory to estimate the transfer rate across thick-thick interfaces. One deficiency in this solution is the fact that the numerical results are too strongly dependent upon the value of the parameter used in the criterion for deciding whether or not a zone is thick. The degree of the deficiency could be reduced by varying the source in zones that are, so to speak, "fairly thick" in such a way that the source gradient at interfaces where the criterion for using the diffusion approximation is nearly met is equal to the diffusion theoretic value. Such measures cannot, however, be expected to provide a totally satisfactory solution to this very difficult problem, which still causes trouble in codes for calculating transport in one-dimensional geometries.

SECTION IV
THE PROGRAM LONG2

The programming details of LONG2 are described in the following paragraphs. The appendix contains a listing of the actual FORTRAN statements.

SUBROUTINE DRAW

Flow of Control

The main function of SUBROUTINE DRAW is to provide SUBROUTINE TRAN2 with representative ray paths, along which radiation transport calculations may be performed for each cycle of a HECTIC problem. Each ray path constructed by DRAW is defined by the indices of the hydro zones pierced by the ray and the segment length of the ray in each of these zones, arranged in an ordered fashion. Since these representative ray paths may be a constant of the HECTIC hydro mesh, SUBROUTINE DRAW has been designed to output this ray-path information onto tape so that only one FORTRAN CALL to SUBROUTINE DRAW need be executed at the beginning of a problem, and none for successive restarts.

The majority of SUBROUTINE DRAW is under control of three nested DO-LOOPS: a θ grid loop, a comb loop, and a ray loop. The θ grid loop is completely defined by four FORTRAN variables (EMU, DMU, DZ1, and DR1) which are input on cards for each grid to be constructed by DRAW. A $\theta = 0$ grid is not allowed as an input grid and is always constructed by DRAW after all the input grids have been constructed.

Construction of a grid is begun by extending the system boundaries to form "ray" boundaries which provide limits beyond which no rays of the grid can intersect the system of hydro zones. Combs of the grid are constructed, and the tracing of each ray of a comb for all combs of the grid is initiated.

The tracing of each ray is always started at the ray's point of closest approach to the vertical axis, which is called the "center" of the

ray, and the tracing is done in two passes called the "upward" pass and the "downward" pass. The upward pass is begun in the section of coding labeled "BEGIN UPWARD TRACE," and the downward pass is begun in the section of coding labeled "BEGIN DOWNWARD TRACE." An attempt is made to first trace the ray in an upward direction, and if possible it is traced in that direction until it leaves the system. (If the center of the ray is at or below a reflecting system bottom boundary, the upward trace cannot be performed, since the ray never enters the system.) After the ray has been traced in an upward fashion through its possible reflection at the top system boundary and successive downward path and exit, control is again returned to the ray's center and the downward pass is performed. If the ray happens to have its center above the top boundary which is transmittive, no upward pass is performed and the first pass is a downward pass. Execution of the downward pass is similar to that of the upward pass.

The upward and downward tracing within each pass is performed by sections of the code labeled "GENERAL CASE OF INCREASING Z" and "GENERAL CASE OF DECREASING Z," respectively. There are several smaller blocks of coding that feed into the above two sections which initialize parameters for the beginning of an upward or downward trace, or after a reflection. The FORTRAN variable IUP indicates whether an upward pass or a downward pass has been performed, and tracing is complete when the ray has been completely followed through the system.

If a ray enters the system, the weight WT associated with that ray is added into the appropriate location of one of the FORTRAN arrays BETAA, BETAR, or BETAB, depending on whether the ray enters the top (above), side (right), or bottom system boundary, respectively. This information is made available to SUBROUTINE TRAN2 for use in calculating the initial radiative energy a ray possesses when it enters the system via one of the system boundaries which have black-body temperature profiles associated with them. Rays constructed in DRAW which intersect the vertical axis have half the weight that would normally be assigned to them since in

TRAN2 transport is performed in two directions along these rays and, in actuality, these rays represent only one possible set of rays in the system.

When the tracing of a ray through a hydro zone is performed, the segment length of the ray in that hydro zone and the index of the hydro zone are stored in temporary storage arrays XB and KXA, respectively.

When the trace of a ray is completed, if two passes are actually performed it is necessary to connect the two separate strings of data in such a way as to have a continuous chain of zone indices or segment lengths portraying the path of the ray through the system. Before the indices and segments are merged into one output storage array buffer (ER ≡ NER) by a packing section of code, each segment length of the path is added to the appropriate location in the XALP array.

As the rays of each grid are traced and the output storage buffer is filled with indices, segments, the weight WT of each ray, and flags to show through which system boundaries the ray enters and exits, the buffer is output onto tape as its contents reach the limit of 1023 words.

After the input grids have been constructed, the construction of the rays of the $\theta = 0$ grid is treated in essentially the same manner as that of the rays of the input grids. One situation in which the $\theta = 0$ grid is exceptional is in the presence of a symmetry plane. When there is a symmetry plane at the bottom, an ordinary ray will have its center above the plane, and its image will be a different ray whose center is below the plane. But a ray that is parallel to the axis is its own image, and it would be redundant to trace it both forward and backward. TRAN2 is apprised of this fact by setting NXIT=-1 in the prologue of rays for which $\theta = 0$ for symmetrical problems.

At the end of DRAW, a terminal record is written on the tape of ray data and control is returned to MAIN of HECTIC.

Input to DRAW

Medium	Variables
CARDS	CLAMDA
	ATHETA(I), I=1,IMAX
	RTHETA(J), J=1,JMAX
	BTHETA(I), I=1,IMAX
	ANGN
	(EMU, DMU, DZ1, DR1), IANG = 1, NANG

Error Termination in DRAW

Type	Indicator	Description of Error
INPUT	S1 = 5.0025	$X(0) > 10^{-10}$
INPUT	S1 = 5.0021	$\cos\theta > .99995$
INPUT	S1 = 5.0022	$\sum_{i=1}^{NANG} DMU > 1.$
INPUT	S1 = 5.0023	FIX(IMAX/DR1 + 0.8) > 100; that is, the numbers of combs per grid is > 100
INPUT	S1 = 5.0024	BCBTAG and BCATAG < 0
PROGRAMMING	S1 = 5.0075	--
PROGRAMMING	S1 = 5.0082	--
PROGRAMMING	S1 = 5.0105	--
PROGRAMMING	S1 = 5.0112	--
PROGRAMMING	S1 = 5.0135	--
PROGRAMMING	S1 = 5.0160	--

Results of DRAW

Medium	Variables	
COMMON	CLAMDA	
	ATHETA(I).	I=1,IMAX
	RTHETA(J),	J=1,JMAX
	BTHETA(I),	I=1,IMAX
	BETAA(I),	I=1,IMAX
	BETAR(J),	J=1,JMAX
	BETAB(I),	I=1,IMAX
	XALP(K),	K=2,KMAX
DRAW TAPE OR DISK	Contains for each ray traced:	
	1) Index of each hydro zone pierced	
	2) Length of ray segment in each pierced zone	
	3) System boundary through which ray enters	
	4) System boundary through which ray exits	
	5) Weight associated with ray	

DRAW Glossary

FORTRAN Label	Report Label	Description
ANGN	N-1	The number of polar angles θ to be used; or, equivalently, the number of grids to be constructed by DRAW. (input on a card).
ATHETA(I)	θ_{out}	The temperature of the black-body system boundary adjacent to cell (I, JMAX) (in eV) (input on cards) (used in TRAN2)
BCATAG	None	Boundary flags. A and B refer to top (above) and bottom system boundaries, respectively. If $\theta < 0$, boundary is a perfect reflector; otherwise, it is transmittive to radiation.
BCBTAG	None	
BETAA(I)	None	$\sum_{\substack{\text{all rays passing} \\ \text{through the system} \\ \text{boundary adjacent to} \\ \text{zone (I, JMAX)}}}$ (weight WT of each ray)
BETAB(I)	None	$\sum_{\substack{\text{all rays passing} \\ \text{through the system} \\ \text{boundary adjacent to} \\ \text{zone (I, 1)}}}$ (weight WT of each ray)
BETAR(J)	None	$\sum_{\substack{\text{all rays passing} \\ \text{through the system} \\ \text{boundary adjacent to} \\ \text{zone (IMAX, J)}}}$ (weight WT of each ray)
BTHETA(I)	θ_{out}	The temperature of the black-body system boundary adjacent to zone (I, 1) (in eV) (input on cards) (used in TRAN2)
CLAMDA	c_λ	Thick-thin zone criterion

		(in number of mean free paths) (input on a card) (used in TRAN2)
COTAN	None	COTθ
DBGPRT	None	The debug print control: $\lvert DBGPRT \rvert \geq 10^{-20}$ means debug print desired; otherwise, debug print not desired.
DELS	t	In tracing a ray, the length (or segment) of a ray in a hydro zone. (in cm)
DELX	None	The projection of t onto a horizontal plane (in cm)
DELZ	None	The projection of t onto the vertical axis (in cm)
DMU	$\Delta\mu$	Δcos θ for a θ grid (input on cards)
DR1	Δy	Radial distance between rays of a θ grid (constant for a grid) (in cm) (input on cards)
DXF	None	1/sin θ
DY(J)	$z_j - z_{j-1}$	Vertical length of hydro zones (I,J), I = 1, IMAX (in cm)
DZ1	Δz^*	Vertical distance between rays of a θ grid (constant for a grid) (in cm) (input on cards)
DZF	None	1/cos θ
EMU	μ	cos θ, the input parameter that specifies a θ grid (input on cards)

ER	None	≡NER. The output storage buffer array for ray data that is to be written on the ray tape
ETA	η	$\sin \theta$
FIOUT	None	Saved temporarily on disk in DRAW
I	None	A running index
IANG	None	Running index of the θ grid DO-LOOP
IB	None	Temporary storage for a subscript
ICOR	None	In tracing a ray, ICOR = 1 means the ray hits the corner of a hydro zone ICOR = 0 means the ray does not hit the corner of a hydro zone
IDNMN	None	In merging starting segments of the upward and downward passes, IDNMN = 201 means starting segments for each pass are not in the same hydro zone IDNMN = 202 means starting segments for each pass are in the same hydro zone
IEND	None	A DO-LOOP limit in the θ = 0 trace
IGWD	None	In tracing rays, the index of the last word loaded into the output storage buffer array ER (which is limited to 1023 words). When the storage array is filled and written onto tape, ER(1) = NER(1) = IGWD
II	None	A running index
II2	None	≡ITOT. The total number of hydro zones a ray penetrates
IIR	None	A running index
IITOT	None	The accumulative number of ray segments generated by DRAW
IM	None	A running index
IMAX	i_{max}	The number of radial hydro zones
INW	None	Temporary storage for a subscript

IQ	None	The number of radial hydro zones that a ray penetrates in traversing the system
IQQ	None	A running index
IRD	None	Running index of the comb DO-LOOP
IRSTRT	None	Index I of the innermost radial hydro zones (I, J), J=1, JMAX intersected by a comb
IRX	L_n	The number of combs in a θ grid
ISEND	None	An error flag used by SUBROUTINE EDIT
ITOP	None	In tracing a ray, ITOP = 1 means the center of a ray lies at or above the plane Y(JMAX) ITOP = 0 means otherwise
ITOT	None	≡II2. (see II2)
ITR	None	In tracing a ray, the radial hydro zone which contains the current ray segment
ITWD	None	A running index
IUP	None	The subscript of the last entry made in the arrays KXA and XB: IUP ≤ 200 on the upward pass IUP > 200 on the downward pass
IUPM	None	A DO-LOOP limit
IUPUP	None	The last value of IUP on the first (upward) pass
IXX	None	Given XTR, IXX is such that RX(IXX) > XTR and RX(IXX-1) ≤ XTR
IZRAY	None	Running index of the ray DO-LOOP
J	None	A running index
JB	None	A running index
JMAX	j_{max}	In tracing a ray, the vertical hydro zone which contains the current ray segment
JZX	None	The numbers of rays in a comb
K	None	The subscript of a hydro zone
KMAX	None	KMAX-1 is the total number of hydro zones in the system

KMAXA	None	KMAX+1
KTR	None	In tracing a ray, the index of the hydro zone which contains the current ray segment
KX1	None	Temporary storage for integers
KXA	None	KXA(IUP) contains the subscript of the hydro zone containing the current segment (XB(IUP)) of the ray being traced
L1	None	A DO-LOOP limit
L2	None	A DO-LOOP limit
L3	None	In summing all ray segment lengths contained in a zone to compute XALP(K), L3 = 1 means summing first pass segments L3 = 2 means summing second pass segments
NANG	N-1	Number of θ grids to be constructed by DRAW
NER	None	≡ ER
NPOINT	None	A pointer used to return control from the ray-packing section of DRAW
NRGP	None	The number of current ray traces in the output storage buffer, ER ≡ NER
NTER	None	NTER = (1/2/3) indicates that a ray enters the system through the (bottom/right/top) boundary
NWD	None	The number of new words to be added to the current accumulation of ray data in the output storage buffer, ER ≡ NER
NXIT	None	NXIT > 0 NXIT = (1/2/3) indicates that a ray leaves the system through the (bottom/right/top) boundary NXIT < 0 NXIT < 0 flags a θ = 0 trace for TRAN2
P	None	Saved temporarily on disk in DRAW
PARIM	None	Temporary storage
PI	π	π
QW	None	Temporary storage

RPT(IRD)	None	The radial coordinate of the "center" (see ZCNTR) of the rays of comb IRD (in cm)
RTHETA(J)	θ_{out}	The temperature of the black-body system boundary adjacent to zone (IMAX,J) (in eV) (input on cards) (used in TRAN2)
RX	None	RX(I), I=1,IQ, is the horizontal projection of the distance along a ray from the "center" (see ZCNTR) of the ray to successive radial zone boundaries intersected by the ray (being the same for all rays in a comb) (in cm)
S1	None	Error flag for EDIT
SDELX SDELZ	None None	When ZCNTR of a ray is not between Y(0) and Y(JMAX), SDELX is the horizontal projection of the distance along the ray from its "center" (see ZCNTR) to the point where it enters the system, and SDELZ is the axial projection (in cm)
SMU	None	$1 - \sum_{i=1,\,IANG} DMU$
TAU(I)	$\pi\left(r_i^2 - r_{i-1}^2\right)$	The area of the annulus defined by radial hydro interval I (in cm^2)
U	None	≡ROSS. Saved temporarily on disk in DRAW
V	None	≡KXA. Saved temporarily on disk in DRAW
WT	$\omega_{\ell,m,n}$	The weight associated with a given ray: If $\theta \neq 0$, $WT = (\Delta\cos\theta\ \Delta Z\ \Delta R\ \sin\theta)(F)$, where $F = 1$ if $IRD \neq 1$ and $F = .5$ if $IRD = 1$ If $\theta = 0$, $WT = 1/2 \sin\theta\ TAU(I)$

WTP	None	Temporary storage, used in the WT calculation for $\theta = 0$ trace
X(I)	r_i	Outer radius of radial hydro zone I (in cm)
XALP(K)	$4\pi A$	$= 2*TAU(I)*\Delta y*\pi/\sum_{\text{all rays passing through zone K}}$ [(length of ray in zone K) *(weight WT of ray)]
XB	None	XB(IUP) contains the segment length in the current zone (KXA(IUP)) of the ray being traced. traced
XTR	None	In tracing a ray, XTR is the horizontal component of the distance along a ray from the "center" (see ZCNTR) of the ray to the beginning of the current ray segment (in cm)
XX2	None	Temporary storage
Y(J)	z_j	The vertical coordinate of the upper boundary of hydro zones (I,J), I=1,IMAX (in cm)
ZCNTR	None	The axial coordinate of the point of closest approach of a ray to the vertical axis, or the "center" of a ray (in cm)
ZEND	None	The axial coordinate of the "center" (see ZCNTR) of ray (IZRAY=JZX) (in cm)
ZSTRT	None	The axial coordinate of the "center" (see ZCNTR) of ray (IZRAY=1) (in cm)
ZTR	None	In tracing a ray, ZTR is the axial coordinate of the beginning of the current ray segment (in cm)
ZXTRA	None	X(IMAX)*cot θ, used in the calculation of ZSTRT and ZEND (in cm)

SUBROUTINE TRAN2

Flow of Control

The main function of SUBROUTINE TRAN2 is to perform transport calculations along the representative ray paths supplied by SUBROUTINE DRAW, extracting and depositing radiative energy from the HECTIC hydro zones according to the temperature and optical depth of each zone.

Upon entering SUBROUTINE TRAN2 from MAIN, the integrated black-body intensity B(K) for each hydro zone is calculated if the zone is considered "active" by HECTIC standards.

Next, the black-body boundary conditions are calculated. The amount of energy E_{SURF} entering the system boundary through zone K from a black-body boundary at temperature θ (ATHETA, RTHETA, or BTHETA in the code) is given by Eq. (133):

$$E_{SURF} = \frac{ac}{4} \theta^4 \alpha \underbrace{\int_{h\nu_1/\theta}^{h\nu_2/\theta} \frac{15}{\pi^4} \frac{u^3 du}{e^u - 1}}_{\text{FUNCTION PLNKUT in the code}}$$

where ac/4 = Stefan's constant

α = the area which zone K presents to the black-body boundary

ν_1, ν_2 = the frequency limits ($h\nu_1 = 0.001$ eV, $h\nu_2 = 10^6$ eV in the monofrequency problem)

h = Planck's constant

The amount of this energy apportioned to each constructed ray (with weight WT_{RAY}) which enters the system through zone K is given by

$$E_{RAY} = \frac{E_{SURF} \cdot WT_{RAY}}{\sum\limits_{\substack{\text{all rays}\\\text{passing from the}\\\text{boundary into zone K}}} WT}$$

In the section of coding prior to the transport calculations, only the quantities E_{SURF}/Σ WT are calculated (in the code, PA,PR1,PB).

Also prior to the transport calculations, the Rosseland mean absorption coefficients are calculated (ROSS(K)) for each hydro zone, and the diffusion flags are set. If a zone is less than CLAMDA (input) mean free paths thick or is inactive, it is considered "optically thin;" otherwise, it is labeled "optically thick" (the flags occupy the ALAMV array).

The two-dimensional transport calculation section is based around three nested loops: the "RAY-TAPE RECORD LOOP," the "RAY LOOP," and the "SEGMENT LOOP." The input storage buffer array (XB ≡ KXB) is filled with a logical record of ray data from the ray tape which contains data from traces of one or more rays constructed in SUBROUTINE DRAW. The "RAY LOOP" is then entered, and initialization for transport begins for each ray trace in the buffer. If the ray traverses a totally inactive region, the transport calculations are skipped and the next ray trace in the buffer is considered.

In general two "passes" or "transports" are performed for each ray trace supplied by DRAW. The first pass (IPASS = 1) is performed by picking up zone indices from the array KXA in an increasing manner, and the second pass (IPASS = 2) is performed "backwards" in a sense to the first pass along the same ray trace supplied by DRAW. If the ray is vertical ($\theta = 0$, NTER = -1), only one pass is performed. For each pass, the rate at which the hydro zones are gaining energy due to radiation is calculated (ER(K)), the rate at which the system is gaining energy through its boundaries is calculated (EBTM, ETOP, ESIDE), the total rate at which the system is gaining energy is calculated (FOO), and the W2 array is updated.

In the "SEGMENT LOOP," if the ray enters the system through a black-body boundary the rate at which energy is transported into the zone containing its first segment is given by

$$E_{RAY} = \begin{pmatrix} PB(I) \\ PRI(J) \\ \text{or} \\ PA(I) \end{pmatrix} * WT_{RAY}$$

In transporting along the trace of a ray, the diffusion flag of the zone in which its next segment lies is checked. If the zone is considered "thick," then all the energy/(DT) transported by the ray is emptied into that zone. As the trace continues, no deposition calculation is performed and the diffusion flags of the zones are checked until a "thin" zone is encountered. Then the rate at which energy is leaving the last "thick" zone is give by

$$E_{RAY} = B(K) \cdot XALP(K) \cdot WT_{RAY}$$

The outside of the system is treated as if it were "thin." For a sequence of segments in "thin" zones, the transport is calculated in the section of coding labeled "NORMAL TRANSPORT CALCULATION" under the equation

$$E_{OUT} = E_{IN}\, e^{-\sigma s} + c_1 c_2 (1 - e^{-\sigma s})\; WT_{RAY}$$

where E_{OUT} = rate at which energy is leaving zone K containing segment length s

E_{IN} = rate at which energy is entering zone K containing segment length s

σ = Rosseland mean absorption coefficient for zone K (ROSS(K))

c_1 = XALP(K)

c_2 = B(K)

WT_{RAY} = weight associated with the ray

s = segment length in zone K

After transport calculations have been performed along all the rays constructed by DRAW, an explicit diffusion calculation is performed across all "thick-thick" zone interfaces.

At the end of SUBROUTINE TRAN2, the ER array is checked to see if radiation transport in time DT will change the internal energy of a hydro zone by more than the allowed amount (SLUG); if it will, DT is reduced in TRAN2 so as to comply with this restriction. Finally, the internal energy of the system, ETH, is updated to include energy changes by radiation transport, and the variables BACC, SACC, and TACC are updated.

The outside loop in TRAN2 is the frequency loop which is closely patterned after the one in SPUTTER (Ref. 5). The multifrequency option has never been tried, however, in any of the radiation subroutines (TDRAD, TRAN2/LONG2, and TRAN2/SHORT2) that have been incorporated into HECTIC.

Input to TRAN2

Results from SUBROUTINE DRAW are transferred to TRAN2. (See the subsection entitled "Results of DRAW.")

Error Termination in TRAN2

Type	Indicator	Description of Error
INPUT	S1 = 7.0225	MERGE ≤ 0 and multifrequency problem
INPUT	S1 = 7.0235	IHNU < 1
INPUT	S1 = 7.0250	Q ≥ 0.6 or Q ≤ 0.4
INPUT	S1 = 7.0320	XA(K) < 0.
EXECUTION/ INPUT	S1 = 7.0522	An indefinite number has been created before the call to DVCHK
INPUT	S1 = 7.0541	There are more than 210 ray traces for an input buffer-load of ray data from DRAW (it is impossible for DRAW to create correctly more than 170 traces in one buffer-load)

INPUT	S1 = 7.0550	In transporting along a ray, NTER is not 1, 2, or 3
INPUT	S1 = 7.0556	NTER = 1 and BCBTAG < 0.
EXECUTION/ INPUT	S1 = 7.0810	An indefinite number has been created before the call to DVCHK
INPUT	S1 = 7.1010	IHNU > NHNU
EXECUTION	S1 = 7.1068	DT < FFB, the minimum time-step control

Results of TRAN2

Medium	Variables
COMMON	(A possible contribution to)DT
	(A contribution to)ETH
	ER(K), K=2,KMAX
	W2(I), I=1,IMAX

TRAN2 Glossary

FORTRAN Label	Report Label	Description
AH	None	In the explicit diffusion calculation, a variable used in the calculation of net diffusion in the horizontal direction
AIX(K)	None	The total internal energy of zone K per unit mass (does not include kinetic energy) (in ergs/gm)
AJ	J	In tracing a ray through a hydro zone of the system, the rate at which energy is being transported into or out of the zone (in ergs/sec)
AJN	J	Similar in function to AJ (in ergs/sec)

ALAMV(K)	None	For the current problem cycle, ALAMV(K) = 1 means zone K is optically thick and active ALAMV(K) = 0 means zone K is optically thin or inactive
AMX(K)	None	The mass of zone K (in gm)
ATHETA(I)	θ_{out}	The temperature of the black-body system boundary adjacent to zone (I,JMAX) (in eV) (input in DRAW)
AV	None	In the explicit diffusion calculation, a variable used in the calculation of net diffusion in the vertical direction
B(K)	$\frac{S}{\sigma}$	In the monofrequency calculation, the integrated black-body intensity associated with zone K (in $\frac{\text{ergs}}{\text{sec-steradian-cm}^2}$)
	None	In the multifrequency calculation, temporary storage
BACC	None	$\int_{t'=0}^{t'=t}$ (rate at which energy is leaving the system bottom boundary due to radiation) dt' (in ergs)
BCATAG	None	Boundary flags. A, B, and R refer to top (above), bottom, and side (right) system boundaries, respectively. If $\theta < 0$, boundary is a perfect reflector; otherwise, it is transmittive to radiation (the side cannot be a reflector)
BCBTAG	None	
BCRTAG	None	
BETA	$\frac{h\nu_1}{\theta}$	In the multifrequency calculation, the lower limit of the current frequency band

BETAA(I)	None	$\sum$(weight of each ray) all rays passing through the system boundary adjacent to zone (I, JMAX) (calculated in DRAW)
BETAB(I)	None	$\sum$(weight of each ray) all rays passing through the system boundary adjacent to zone (I, 1) (calculated in DRAW)
BETAP	$\frac{h\nu_2}{\theta}$	In the multifrequency calculation, the upper limit of the current frequency band
BETAR(J)	None	$\sum$(weight of each ray) all rays passing through the system boundary adjacent to zone (IMAX, J) (calculated in DRAW)
BTHETA(I)	θ_{out}	The temperature of the black-body system boundary adjacent to zone (I, 1) (in eV) (input in DRAW)
CLAMDA	c_λ	Thick-thin zone criterion: If an active zone K has vertical and radial dimensions ≥ CLAMDA/ROSS(K), then it is considered optically thick; otherwise, it is considered optically thin (in number of mean free paths) (input in DRAW)
DBGPRT	None	The debug print control: $\lvert DBGPRT \rvert \geq 10^{-20}$ means debug desired; otherwise, debug print not desired
DE	None	In calculating DT in TRAN2, the amount of energy per unit mass which a zone will lose or gain in time DT due to radiation (in ergs/gm)

DERH	$\dot{E}_d$	In the explicit diffusion calculation, the net diffusion rate in a horizontal directio . between zones (in ergs/sec)
DERV	$\dot{E}_d$	In the explicit diffusion calculation, the net diffusion rate in a vertical direction between zones (in ergs/sec)
DFB	None	In the multifrequency calculation, $$\frac{15}{\pi^4}\int_{BETA}^{BETAP} \frac{u^3}{e^u - 1}\, du$$
DHNU	None	In the multifrequency calculation, the width of the current frequency band
DT	None	Time step (which can be controlled by TRAN2) (in sec)
DTEMP	None	Temporary storage
DX(I)	$r_i - r_{i-1}$	Radial zone length of zones (I, J), J=1, JMAX
DY(J)	$z_j - z_{j-1}$	Vertical zone length of zones (I, J), I=1, IMAX
E	None	In the temperature iteration calculation, the total internal energy per unit mass updated by radiation heating only (in ergs/gm)
EBTM	None	The rate at which energy is entering the system through the system bottom boundary in time DT due to radiation (in ergs/sec)
EMB1	None	In the multifrequency calculation, temporary storage
EMB2	None	Same as above
ER(K)	None	The rate at which zone K is gaining energy due to radiation in time DT (in ergs/sec)

	None	In the multifrequency calculation, temporary storage
ESIDE	None	The rate at which energy is entering the system through the system side boundary in time DT due to radiation (in ergs/sec)
ESS	None	Temporary storage used in the "normal transport calculation"
ETH	None	The total energy in the system at time T (in ergs)
ETOP	None	The rate at which energy is entering the system through the system top boundary in time DT due to radiation (in ergs/sec)
FACTOR	None	Temporary storage used in the calculation of ROSS(K) and PLANCK(K)
FFB	None	The minimum time-step allowed by HECTIC (in sec)
FIOUT	None	= OLDTH Saved temporarily on disk in DRAW and restored at the end of TRAN2
FOO	None	The amount of energy gain by the system in time DT due to radiation (in ergs)
GG	None	A dummy argument in the call to ES in TRAN2
HNU	$h\nu_1$	In the multifrequency calculation, the lower frequency of the current frequency band (in eV)
HNU4	None	= HNU^4.
HNUP	$h\nu_2$	In the multifrequency calculation, the upper frequency of the current frequency band. (in eV)
HNUP4	None	= $HNUP^4$.
I	None	A running index.

IACT	None	An activity flag: IACT = 1 means all the zones traversed by a ray constructed in DRAW are inactive, and no transport is performed IACT = 2 means some or all of the zones traversed by a ray constructed in DRAW are active, and transport is performed
IDY	None	Running index of the segment loop for a ray trace
IGRP	None	Running index of the ray-trace loop
IGWD	None	In tracing a ray within the ray-tape record loop, a pointer to the end of the ray data in the input storage array XB (≡KXB) for the current ray
IHNU	None	In the multifrequency calculation, the index of the current frequency band
II	None	A running index
IMAX	i_{max}	The number of radial hydro zones
IPASS	None	In transporting along a ray, IPASS = 1 means transport is being performed along a ray in a direction specified by the increasing order of zone indices specified in the KXB array IPASS = 2 means transport is in the opposite direction to that of IPASS = 1
ISEND	None	An error flag used by SUBROUTINE EDIT
ITAG	None	Flag for the temperature iteration calculation: = 0 means perform temperature iteration ≠ 0 means do not perform temperature iteration
ITOT	None	In transporting along a ray, the number of hydro zones that a ray penetrates in one pass through the system
ITR	None	In transporting along a ray in the ray segment loop, the pointer to the location in the KXA (≡XA) array containing the current zone index (ITR+400 points to the location in the XA(≡KXA) array containing the corresponding ray segment length)

J	None	A running index
JMAX	j_{max}	The number of vertical hydro zones
K	None	A running index
KDMY	None	Argument returned by the calls to FUNCTION DVCHK: = 1 means an error = 2 means no error
KFIT	None	An array constructed by HECTIC such that: JMR(KFIT(K),2) = 1 means zone K is active JMR(KFIT(K),2) ≠ 1 means zone K is inactive
KMAX	None	KMAX-1 is the total number of hydro zones in the system
KMAXA	None	KMAX+1
KOLD	None	In transporting along a ray in the ray-segment loop, the index of the zone which contains the previous ray segment
KPP	None	In the explicit diffusion calculation, the index of the zone above zone K
KSTRT	None	In transporting along a ray, the index of the first hydro zone which the ray intersects upon entering the system
KT	None	A running index
KXA	None	≡XA. In transporting along a ray, the storage array which contains the ray data for the current ray. KXA(K) contains the index of the hydro zone pierced by the ray (the corresponding ray segment length is contained in XA(K+400))
KXB	None	≡XB. The input storage buffer array for DRAW ray data (contains traces of one or more rays)
M	None	Not used
MERGE	None	In the multifrequency calculation, a variable controlling the merging of frequency bands

MFTAG	None	Multifrequency flag: = 0 means monofrequency problem ≠ 0 means multifrequency problem
N	None	Not used
NC	None	Integer value of cycle number
NGRP	None	The number of ray traces in the current buffer-load of ray data (XB ≡ KXB)
NGWD	None	The number of words in the current buffer-load of ray data (XB ≡ KXB)
NHNU	None	In the multifrequency calculation, the total number of frequency bands
NTER	None	NTER = (1/2/3) indicates that a ray "enters" (depending on the pass) the system through the (bottom/side/top) boundary
NVEZ	None	In the temperature iteration calculation, the temperature iteration counter
NXIT	None	NXIT > 0 NXIT = (1/2/3) indicates that a ray "leaves" (depending on the pass) the system through the (bottom/side/top) boundary NXIT < 0 means $\theta = 0$ trace
NY	None	Not used
OLDTH	None	≡ FIOUT In the temperature iteration calculation, temporary storage
P	None	≡PLANCK. Saved temporarily on disk in DRAW and restored at the end of TRAN2
PA(I)	None	The rate at which energy is entering zone (I,JMAX) along a ray from the adjacent black-body system boundary at temperature ATHETA(I), divided by the weight WT of the ray (in ergs/sec)

PB(I)	None	The rate at which energy is entering zone (I, 1) along a ray from the adjacent black-body system boundary at temperature BTHETA(I), divided by the weight WT of the ray (in ergs/sec)
PI	π	π
PLANCK(K)	None	In transporting along a ray through hydro zone K, the rate at which energy is entering zone K from the zone previously traversed (in ergs/sec) Also, the PLANCK mean opacity across the merged frequency band (in 1/cm)
PRI(J)	None	The rate at which energy is entering zone (IMAX, J) along a ray from the adjacent black-body system boundary at temperature RTHETA(J), divided by the weight WT of the ray (in ergs/sec)
PUR	None	Not used
PUZ	None	Not used
Q	None	In the calculation of DT in TRAN2, the amount by which the internal energy per unit mass of a zone is allowed to change due to radiation transport (in ergs/gm) Also, in the multifrequency calculation, a parameter used in the merging of frequency bands
ROSS(K)	σ	The Rosseland mean absorption coefficient for zone K (in 1/cm)
RPTAG	None	Absorption coefficient flag: $\neq 0$ means set the Planck mean absorption coefficient equal to the Rosseland mean absorption coefficient = 0 means keep both absorption coefficients distinct

RTHETA(J)	θ_{out}	The temperature of the black-body system boundary adjacent to zone (IMAX, J) (in eV) (input in DRAW)
RUR	None	Not used
RUZ	None	Not used
S1	None	Error flag for EDIT
SACC	None	$\int_{t'=0}^{t'=t}$ (rate at which energy is leaving the system bottom boundary due to radiation) dt' (in ergs)
SIGS	None	Temporary storage
SLUG	None	In the calculation fo DT in TRAN2, the fraction by which the internal energy per unit mass of a zone is allowed to change due to radiation transport
SV	None	Temporary storage
T4	None	In merging frequency bands, = THETA(K)^4 (in eV^4)
TACC	None	$\int_{t'=0}^{t'=t}$ (rate at which energy is leaving the system top boundary due to radiation) dt' (in ergs)
TAU(I)	$\pi(r_i^2 - r_{i-1}^2)$	The area of the annulus defined by radial hydro interval I (in cm^2)
TEMP(1)	None	In changing DT in TRAN2, the computed fraction by which DT is reduced to comply with the restrictive variable SLUG
THETA(K)	θ	Temperature of zone K (in eV)

THTAMX	None	In the multifrequency calculation, the highest temperature in the system used in deciding whether to merge frequency bands (in eV)
U	None	≡ ROSS. Saved temporarily on disk in DRAW and restored at the end of TRAN2
V	None	≡ KXA ≡ XA. Saved temporarily on disk in DRAW and restored at the end of TRAN2
VEZ	None	Not used
W2(I)	None	The rate at which energy is leaving the system top boundary adjacent to zone (I, JMAX) due to radiation, divided by TAU(I) (in ergs/cm^2 sec)
WT	$\omega_{\ell,m,n}$	The weight associated with a given ray (see "DRAW Glossary").
X(I)	r_i	Outer radius of radial hydro zone I (in cm)
XA	None	≡ KXA (see KXA)
XALP(K)	$4\pi A$	$= 2*TAU(I)*\Delta y*\pi/\sum_{\text{all rays passing through zone K}}$ (length of ray in zone K) *(weight WT of ray)
XB	None	≡ KXB. (see KXB)
Y(J)	z_j	The vertical coordinate of the upper boundary of hydro zones (I, J), I=1, IMAX (in cm)

SECTION V

THE VIEW-FACTOR METHOD

INTRODUCTION

The view-factor method is proposed as an economical method for calculating radiative transport of energy in situations where diffusion theory is inapplicable and higher-order methods are unfeasible. The problem of distributing rays does not exist because the concept of rays as characteristics is not used. A procedure for interpolating the source strength between zone centers that is consistent with diffusion theory is a feature of the method.

Let Z be a hydro zone in HECTIC and let B_b, b = 1,2,3,4, denote its boundaries. The sphere of directions of rays passing through any point of the system is partitioned into bundles, called "cones" and denoted by C_a, a = 1,2,..., A. Currently, the assignment of directions to cones at various points is the same for all points of the system relative to the local coordinate system. This means that if at point P a direction with axial projection μ and azimuth ϕ is in cone C_a, then at any other point, say Q, the direction with the same axial projection and azimuth is also in cone C_a. Cones are to be constructed so that the rays of a cone are all rising or all descending and all approaching or all receding from the axis as they pass through the vertex. The function of this restriction is to ensure that if one ray in cone C_a with vertex P on B_b enters Z at P, then all rays of that cone do the same. The specification of cones used in SHORT2 is discussed in the next subsection.

The basic construct of the view-factor method is an object called a "fan." The fan $F_{a,b}$ consists of all rays that intersect boundary B_b of zone Z and belong to the cone C_a with vertex at the point of intersection. If fan $F_{a,b}$ leaves zone Z through boundary B_b and fan $F_{a',b'}$ enters zone Z

through boundary $B_{b'}$, then the view factor $\eta_{a,b,a',b'}$ is, roughly, the ratio of the number of rays in fan $F_{a',b'}$ to the number that are in both fans. (Rays that leave and reenter zone Z while passing from $B_{b'}$ to B_b are not counted.) The meaning of "number of rays in fan" is straightforward: the number of rays passing through an element of surface dS with directions lying in an infinitesimal cone of solid angle $d\Omega$ that makes an angle θ with the normal to the surface is $N_o \cos\theta \, d\Omega \, dS$, where N_o is the density of rays, which is assumed to be constant over the system.

To illustrate the concepts "fan" and "view factor," let Z be a HECTIC hydro zone consisting of those points of the system with radii lying in the interval $r_1 < r < r_2$ and aptitudes lying in the interval $z_1 \leq z < z_2$. Let B_b and $B_{b'}$ denote, respectively, the inner and outer boundaries of Z. For any point P on the cylinder $r = r_1$, let the cone C_a with vertex at P consist of those rays through P with axial projections in the range $\mu_1 \geq \mu \geq \mu_2$ that intersect the cylinder $r = \rho_2$ but not the cylinder $r = \rho_1$, where $\rho_1 \leq \rho_2 \leq r$, and that are going inward as they pass through P. Then the fan $F_{a,b}$ consists of all rays that intersect B_b going inward, that have axial projection in the range $\mu_1 \geq \mu \geq \mu_2$, and that intersect the cylinder $r = \rho_2$ but not the cylinder $r = \rho_1$. The number of rays in fan $F_{a,b}$ is

$$N = 2\pi r_1 N_o \int_{\mu_1}^{\mu_2} \int_{\phi_1}^{\phi_2} (1 - \mu^2)^{\frac{1}{2}} \cos\phi \, d\phi d\mu$$

where $r_1 \sin\phi_k = \rho_k$. For any point P' on the cylinder $r = r_2$, let the cone $C_{a'}$ with vertex at P' consist of those rays through P' with axial projections in the range $\mu_1 \geq \mu \geq \mu_2$ that intersect the cylinder $r = \rho_2'$ but not the cylinder $r = \rho_1'$, where $\rho_1 \leq \rho_1' \leq \rho_2 \leq \rho_2' \leq r_2$, and that are going inward as they pass through P'. The fan $F_{a',b'}$ consists of all rays that intersect $B_{b'}$ going inward, that have axial projection in the range $\mu_1 \geq \mu \geq \mu_2$, and that intersect the cylinder $r = \rho_2'$ but not the cylinder $r = \rho_1'$. Let N' be the number of rays in fan $F_{a',b'}$ and let M be the number of rays common to both fans, i.e., the number of rays that intersect both B_b and $B_{b'}$ going

inward, that have axial projections in the range $\mu_1 \geq \mu \geq \mu_2$, and that intersect the cylinder $r = \rho_2$ but not the cylinder $r = \rho_1'$. Then $\eta_{a,b,a',b'} = M/N'$ is the view factor connecting the two fans.

Two properties of view factors follow from the definition:

$$\sum_{a,b} \eta_{a,b;\ a',b'} = 1 \tag{135}$$

and

$$\sum_{a',b'} \eta_{a,b;\ a',b'}\ k_{a',b'} = k_{a,b} \tag{136}$$

where $k_{a,b}$ is the number of rays in fan $F_{a,b}$.

Suppose that zone Z is filled with material having the absorption coefficient σ, and let $J_{a,b}$ be the amount of radiant energy crossing boundary B_b along rays in fan $F_{a,b}$. Then the view-factor method estimates $J_{a,b}$ using the formula

$$J_{a,b} = e^{-\sigma t_{a,b}} \sum_{a',b'} \eta_{a,b;\ a',b'}\ J_{a',b'} + S_{a,b} \tag{137}$$

where $S_{a,b}$ is the rate at which energy radiated in zone Z leaves through boundary B_b along rays in fan $F_{a,b}$, and $t_{a,b}$ is an average distance traveled by rays in fan $F_{a,b}$ in traversing zone Z. The rate of energy deposition in zone Z is then calculated to be

$$\dot{E} = \sum_{a,b} \pm J_{a,b} \tag{138}$$

where the + sign is used if fan $F_{a,b}$ enters zone Z through boundary B_b and the - sign is used if it leaves.

It will be shown later that when $J_{a,b}$ is given for incoming fans associated with the outer boundary of the system, then all others may be determined by a systematic application of Eq. (137).

Some of the principal advantages of the method are as follows:

1. The equations can be solved by a straightforward, noniterative method.
2. The coefficients may be chosen so that the diffusion limit is approached in a continuous, natural manner.
3. The coefficients are all positive. Rapid variation of intensity with direction causes no problems, and negative fluxes cannot occur.

While the view-factor method allows much latitude in the assignment of directions to cones, a simple, rather uneconomical scheme has been chosen for the first version of SHORT2 described here. For some fixed positive integer M, the sphere of directions at each point P of the system is divided into cones $C_{\ell,m}$, $1 \leq \ell, m \leq 2M$, consisting of those directions $\vec{\Omega}$ for which

$$\pi \frac{\ell - 1}{2M} \leq \phi \leq \frac{\pi \ell}{2M} \tag{139}$$

$$\cos \pi \frac{m-1}{2M} \geq \mu \geq \cos \frac{\pi m}{2M}$$

where ϕ is the azimuth and μ the axial projection of the ray passing through P in direction $\vec{\Omega}$. As in the long-characteristic method, the hemisphere of directions of negative azimuth is ignored. The main disadvantage to this type of subdivision is its lack of uniformity; the cones which are more or less vertical have smaller solid angles than do the horizontal zontal ones.

A scheme due to Carlson (Ref. 6) that yields a set of cones of equal solid angle is as follows. With M as before, let $1 = \mu_o > \mu_1 \cdots > \mu_M = 0$, and let

$$\mu_{m-1} - \mu_m = m(1 - \mu_1) \tag{140}$$

Thus

$$\begin{aligned} 1 &= \mu_o - \mu_M \\ &= \sum_{m=1}^{M} \mu_{m-1} - \mu_m \\ &= (1 - \mu_1)\sum_{m=1}^{M} m \\ &= \frac{1}{2} M(M + 1)(1 - \mu_1) \end{aligned}$$

or

$$1 - \mu_1 = \frac{2}{M(M + 1)} \tag{141}$$

For $M < m \leq 2M$, let

$$\mu_{2M-m} = -\mu_m \tag{142}$$

Then, for $\ell \leq 2m \leq 2M$, let

$$\phi_{\ell,m} = \frac{\pi \ell}{2m} \tag{143}$$

and define cone $C_{\ell,m}$ to be those directions $\vec{\Omega}$ for which

$$\begin{gathered} \phi_{\ell-1,m} \leq \phi \leq \phi_{\ell,m} \\ \mu_m \leq \mu \leq \mu_{m-1} \end{gathered} \tag{144}$$

If $M < m \leq 2M$, let $m' = 2M - m + 1$; and for $\ell \leq 2m'$, define cone $C_{\ell,m}$ to be the reflection of $C_{\ell,m'}$ in a horizontal plane through P. Thus,

$$\phi_{\ell-1,m'} \leq \phi \leq \phi_{\ell,m'} \tag{145}$$

$$\mu_m \leq \mu \leq \mu_{m-1}$$

These cones all subtend the same solid angle because

$$\Delta\vec{\Omega} = \Delta\mu\Delta\phi = m(1 - \mu_1)\frac{\pi}{2m} = \frac{\pi}{M(M+1)}$$

for $m \leq M$ and similarly for $m > M$. Figure 4 shows how the octant $0 \leq \mu \leq 1$, $0 \leq \phi \leq \frac{1}{2}\pi$ is subdivided when $M = 4$ in a projection in which meridians, ϕ = const, become concurrent lines and parallels, μ = const, become parallel lines.

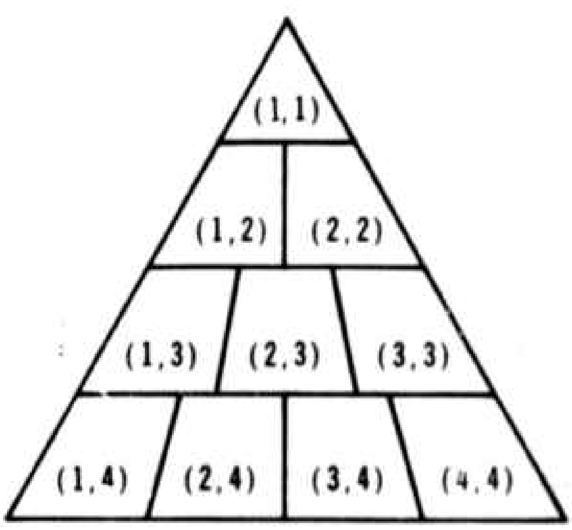


Figure 4. Subdivision of an Octant of the Sphere of Directions

THE MODIFIED DEFINITION OF FAN

The definition of fan given previously is precise but more restrictive than desirable or necessary. Basically, (1) a fan is a set of rays that cross some associated zone boundary in directions that, at the point of crossing, have roughly the same axial projection and azimuth, and (2) fans associated with a given zone boundary are nonoverlapping. The particular method used in SHORT2 for assigning rays to fans is a very slight modification of the one already described. It leads to fewer terms in the summation of Eq. (137) and simplifies the problem of estimating view factors. Only those fans associated with horizontal boundaries are affected. The fan F associated with cone $C_{\ell,m}$ and boundary $BA_{i,j}$ consists of those rays which intersect $BA_{i,j}$ and which cross the cylinder $r = r_i$ in cone $C_{\ell,m}$. Since each ray that penetrates that cylinder crosses it twice, there is an ambiguity to be resolved; if $C_{\ell,m}$ points $\begin{pmatrix}\text{inward}\\ \text{outward}\end{pmatrix}$, then rays of F cross $BA_{i,j}$ while going $\begin{pmatrix}\text{inward}\\ \text{outward}\end{pmatrix}$. The fan associated with vertical zone boundary $BR_{i,j}$ and cone $C_{\ell,m}$ is still the set containing any ray that intersects BR at a point where its direction is in cone $C_{\ell,m}$.

It will prove convenient to have a more compact notation available for the ensuing considerations. Let q denote an index combination for fans, and let F_q be the fan associated with cone

$$C_q = C_{\ell_q,m_q}$$

and boundary

$$B_q = BX^q_{i_q,j_q} ,$$

where X^q = A or R. ($BA_{i,0} \equiv BB_{i,1}$.) Let Z^i_q denote the zone that the rays of fan F_q enter as they cross B_q, and let Z^o_q denote the zone they leave. Equation (137) in this notation, becomes

$$J_q = e^{-\sigma t_q} \sum_{q'} \eta_{q,q'} J_{q'} + S_q \qquad (146)$$

where J_q is the rate of transfer of radiant energy across boundary B_q along rays of fan F_q, and S_q is the rate at which energy radiated in zone Z_q^o crosses boundary B_q along rays of fan F_q.

<u>The View-Factor Integral</u>

Let $k_{q'}$ denote the number of rays in fan $F_{q'}$, assuming that the density of rays is unity. Then

$$k_{q'} = \iint_{F_{q'}} \vec{\Omega} \cdot \vec{n} \, d\Omega \, dS \qquad (147)$$

where the surface integral extends over $B_{q'}$ and for each point on $B_{q'}$, the direction integral extends over directions in cone $C_{q'}$ and $\vec{n}$ is a unit vector normal to $B_{q'}$. If F_q is another fan and if $Z^i_{q'} = Z^o_q$ (i.e., if the zone that $F_{q'}$ enters through $B_{q'}$ is the zone F_q leaves through $B_{q'}$), then

$$\eta_{q,q'} = \frac{1}{k_{q'}} \iint_{F_q \cap F_{q'}} \vec{\Omega} \cdot \vec{n} \, d\Omega \, dS \qquad (148)$$

where, for each point P on $B_{q'}$, the direction integral now extends over directions of rays through P that are common to both fans.

<u>Estimation of the View Factors</u>

In principle, the view factors can be calculated using Eq. (148). Such a procedure involves performing a four-fold integral, two of which are not at all trivial. Evaluating such integrals for each pair of fans would consume much more time than can be justified by the application for which the

results are intended. The current version of the DRAW subroutine of SHORT2 estimates view factors on the assumption that all directions $\vec{\Omega}$ in cone $C_{\ell,m}$ have the same projection $\mu_{m-\frac{1}{2}}$ on the axis of the system, where $\mu_m < \mu_{m-\frac{1}{2}} < \mu_{m-1}$, which eliminates one of the two troublesome integrals. The problem of how to optimize the choice of $\mu_{m-\frac{1}{2}}$ has not yet received any attention. The currently used choice is

$$\mu_{m-\frac{1}{2}} = \cos \frac{2m - 1}{4M} \pi$$

which is the cosine of the polar angle half-way between the polar angles whose cosines are μ_m and μ_{m-1}. Equations (147) and (148) are thus replaced by

$$k_q = \delta_q \int_{B_q} \int_{\phi_{\ell-1}}^{\phi_\ell} \vec{\Omega} \cdot \vec{n} \, d\phi \, dS \tag{149}$$

where $\delta_q = \mu_{m_q - 1} - \mu_{m_q}$ and

$$\eta_{q,q'} = \frac{\delta_{q'}}{k_{q'}} \int_{B_{q'}} \int_{\alpha(P)}^{\omega(P)} \vec{\Omega} \cdot \vec{n} \, d\phi \, dS \tag{150}$$

where $\begin{pmatrix}\omega(P)\\ \alpha(P)\end{pmatrix}$ is the $\begin{pmatrix}\text{upper}\\ \text{lower}\end{pmatrix}$ limit on the azimuths of rays through P that have axial projection $\mu_{m-\frac{1}{2}}$ and belong to both fans.

If F_q is the fan associated with cone $C_{\ell,m}$ and a vertical zone boundary of radius r and axial height h, then

$$k_q = 2\pi r h \, \delta_q (1 - \mu^2_{m-\frac{1}{2}})^{\frac{1}{2}} (\sin \phi_\ell - \sin \phi_{\ell-1}) \tag{151}$$

On the other hand, let the zone boundary associated with F_q be a horizontal one with inner radius r_{i-1} and outer radius r_i. As a point passes through the system along a ray with direction $\vec{\Omega}$, the distance r of the point from

the axis, the azimuth ϕ of $\vec{\Omega}$, and the distance a of the ray from the axis satisfy the relation $r \sin \phi = a$. (See Fig. 5.) It follows that the azimuth ϕ of a ray of F_q will range between $\arcsin (r_i \sin \phi_{\ell-1})/r$ and $\arcsin (r_i \sin \phi_\ell)/r \wedge + 1$ at a distance r from the axis. (Here $a \wedge b$ is

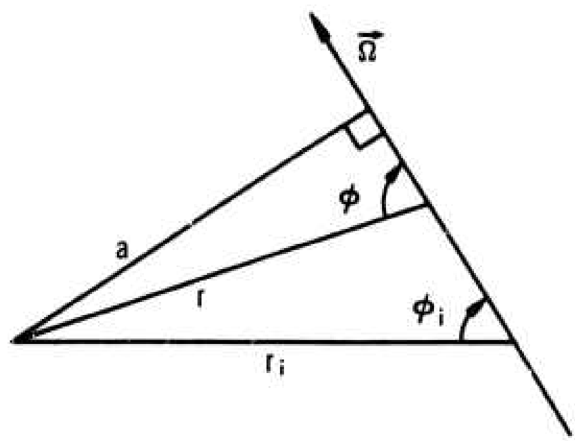


Figure 5. The Dependence of ϕ on r

the lesser and $a \vee b$ is the greater of the two quantities a and b.) It is assumed that $r \geq r_i \sin \phi_{\ell-1}$, for otherwise no ray of F_q will come within a distance r of the axis. The appropriate form for k_q is thus

$$k_q = \delta_q \int_{r_{i-1} \vee r_i \sin\phi_{\ell-1}}^{r_i} 2 \pi r \mu_{m-\frac{1}{2}} \left[\arcsin \left(\frac{r_i \sin\phi_\ell}{r} \vee 1 \right) - \arcsin \frac{r_i \sin\phi_{\ell-1}}{r} \right] dr \tag{152}$$

assuming that $\phi_\ell \leq (1/2)\pi$. When $\phi_\ell > (1/2)\pi$, the equation is still correct provided that the appropriate branch of the arcsine is used. In order to write a formula for k_q in terms of elementary transcendental functions, let $a_\ell = r_i \sin \phi_\ell$ and

$$f_\ell(r) = \int_0^r r' \arcsin \frac{a_\ell}{r'}\, dr' = \frac{r^2}{2} \arcsin \frac{a_\ell}{r} + \frac{a_\ell}{2}\sqrt{r^2 - a_\ell^2}$$

Then

$$k_q = 2\pi\mu_{m-\frac{1}{2}}\, \delta_q \left[f_\ell(r_i) - f_\ell(r_{i-1}) - f_{\ell-1}(r_i) + f_{\ell-1}(r_{i-1})\right]$$

when $r_{i-1} > a_\ell$ and

$$k_q = 2\pi\mu_{m-\frac{1}{2}}\, \delta_q \left[f_\ell(r_i) - \frac{\pi}{4}(r_{i-1} \vee a_{\ell-1})^2 - f_{\ell-1}(r_i) + f_{\ell-1}(r_{i-1} \vee a_{\ell-1})\right]$$

when $r_{i-1} \leq a_\ell$.

As is true of many geometrical problems, the calculation of the coefficients η requires the consideration of many special cases. Let $F_{q'}$ and F_q be a pair of fans for which $Z^i_{q'} = Z^o_q = Z_{i,j}$. Note that $\eta_{q,q'} = 0$ unless $m_q = m_{q'}$, because μ does not change as a point travels along a ray. Consequently, let m be given and assume that $m_q = m = m_{q'}$. Also, denote $\mu_{m-\frac{1}{2}}$ simply by μ and $\phi_{\ell,m}$ by ϕ_ℓ. Further, let $p = \phi_\ell - \phi_{\ell-1}$ so that $\phi_\ell = \ell p$ and $\ell_{max}\, p = \pi$, where ℓ_{max} denotes the maximum value of ℓ. Denote ℓ_q simply by ℓ and $\ell_{q'}$ by ℓ'. Let $h = z_j - z_{j-1}$.

Consider first those cases in which the rays of fan $F_{q'}$ enter zone $Z_{i,j}$ through its outer curved boundary $BR_{i,j}$. Then the integral for the view factor $\eta_{q,q'}$ has the form

$$\eta_{q,q'} = \frac{2\pi r_i (1-\mu^2)^{\frac{1}{2}} \delta_q}{k_{q'}} \int_0^h \int_{\alpha(\zeta)}^{\omega(\zeta)} \cos\phi \, d\phi \, d\zeta \tag{153}$$

where ζ refers to the distance between the point P on $BR_{i,j}$ and the top edge of $BR_{i,j}$ and $\begin{pmatrix}\omega(\zeta)\\ \alpha(\zeta)\end{pmatrix}$ is the $\begin{pmatrix}\text{upper}\\ \text{lower}\end{pmatrix}$ bound on the azimuths of rays through P belonging to both fans. Thus, one obtains

$$\eta_{q,q'} = \frac{2\pi r_i (1-\mu^2)^{\frac{1}{2}} \delta_{q'}}{k_{q'}} \int_0^h \left[\sin \omega(\zeta) - \sin \alpha(\zeta)\right] d\zeta \qquad (154)$$

Outside-to-Inside. Let the rays common to fans F_q and $F_{q'}$ enter $Z_{i,j}$ through the outside boundary and leave through the inside boundary. Rays belonging to fan F_q leave $Z_{i,j}$ at $r = r_{i-1}$ with azimuths in the range $(\ell - 1)p \leq \phi \leq \ell p$, which means that at $r = r_i$, where they enter, their azimuths fall in the range

$$\arcsin \frac{r_{i-1} \sin(\ell - 1)p}{r_i} \leq \phi \leq \arcsin \frac{r_{i-1} \sin \ell p}{r_i}$$

(See Fig. 6.) Consequently, the azimuths of rays common to F_q and $F_{q'}$ fall in the range

$$(\ell' - 1)p \vee \arcsin \frac{r_{i-1} \sin(\ell - 1)p}{r_i} \leq \phi \leq \ell' p \wedge \arcsin \frac{r_{i-1} \sin \ell p}{r_i}$$

It remains to determine which rays in $F_{q'}$ leave zone $Z_{i,j}$ through the inside boundary. Assume that $\mu > 0$ so that the rays have an upward component, and let

$$\tau = \tan \arccos \mu = (1 - \mu^2)^{\frac{1}{2}}/\mu$$

which is the tangent of the angle between the ray and the axis. Then $\tau\zeta$ will be the horizontal distance that a point goes when it rises by an amount ζ along a ray having an axial projection of μ.

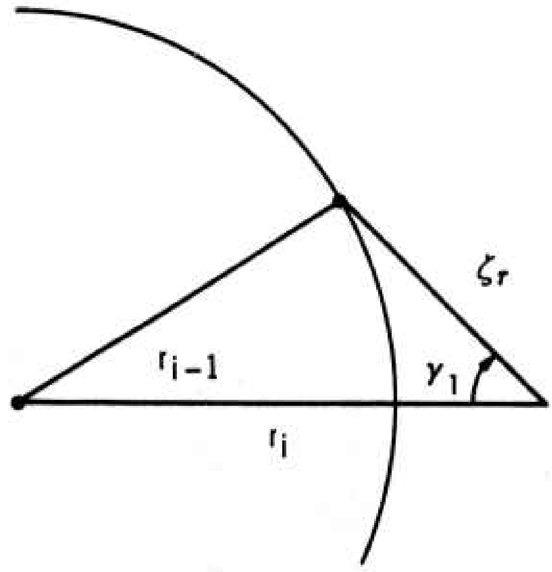


Figure 6. Determination of γ_1

Therefore, a ray of fan $F_{q'}$ that enters zone $Z_{i,j}$ through $BR_{i,j}$ at a point ζ below the top will leave through the inside boundary $r = r_{i-1}$ provided that

$$\cos \phi \geq \frac{r_i^2 + \zeta^2\tau^2 - r_{i-1}^2}{2\, r_i\, \zeta\tau}$$

where ϕ is its azimuth at the outside boundary $r - r_i$, assuming that

$$(r_i - r_{i-1})^2 < \zeta^2 \ \tau^2 < r_i^2 - r_{i-1}^2$$

If $\zeta\ \tau < r_i - r_{i-1}$, then the ray will leave through the top or outside regardless of its incoming azimuth; and if $\zeta\ \tau > (r_i^2 - r_{i-1}^2)^{\frac{1}{2}}$, then it will leave through the inside whenever $\sin \phi < r_{i-1}/r_i$. All this can be summarized by saying that the ray through $BR_{i,j}$ at a point ζ below the top will leave through the inside only if $\phi < \gamma_1(\zeta)$, where

$$\gamma_1(\zeta) = \begin{cases} 0, & \text{if } \zeta\tau < r_i - r_{i-1} \\ \arcsin r_{i-1}/r_i, & \text{if } \zeta^2\tau^2 > r_i^2 - r_{i-1}^2 \\ \arccos \dfrac{r_i^2 + \zeta^2\tau^2 - r_{i-1}^2}{2 r_i \zeta \tau}, & \text{otherwise} \end{cases} \tag{155}$$

It follows that

$$\eta_{q,q'} = \frac{2\pi r_i (1-\mu^2)^{\frac{1}{2}}}{k_{q'}} \int_0^h [\sin\omega(\zeta) - \sin\alpha(\zeta)] \vee 0 \, d\zeta \tag{156}$$

where

$$\omega(\zeta) = \ell' p \wedge \gamma_1(\zeta) \wedge \arcsin \frac{r_{i-1} \sin \ell p}{r_i}$$

$$\alpha(\zeta) = (\ell'-1) p \vee \arcsin \frac{r_{i-1} \sin (\ell - 1)p}{r_i} \tag{157}$$

The part, $\vee 0$, of Eq. (156) is required when $\omega(\zeta) < \alpha(\zeta)$, which will occur, for example, when $\gamma_1(\zeta) < (\ell' - 1)p$, which is the case for small ζ when $\ell' \geq 2$.

<u>Outside-to-Top</u>. Next, let the rays common to fans F_q and $F_{q'}$ enter $Z_{i,j}$ through the outside boundary and leave through the top. This case may be divided into two subcases, depending on whether the rays are traveling inward ($\ell p \leq \pi/2$) or outward ($\ell p > \pi/2$) when they emerge. In either subcase, the azimuths of rays in fan F_q will lie in the range $(\ell - 1)p \leq \phi \leq \ell p$ at $r = r_i$; but in the latter subcase, this will be the situation as they pass out of the cylinder $r = r_i$. It is seen from figure 7 that the incoming and outgoing azimuths are supplementary and hence that

the range of azimuths of rays in F_q where they enter the cylinder $r = r_i$ is, for $\ell p > \frac{1}{2}\pi$,

$$(\ell_{max} - \ell)p \leq \phi \leq [\ell_{max} - (\ell - 1)]\,p$$

Therefore, $\eta_{q,q'} = 0$ unless $\ell' = \ell$ or $\ell' = \ell_{max} - (\ell - 1)$. Equation (156) is still valid for $\eta_{q,q'}$, but the formulas for $\omega(\zeta)$ and $\alpha(\zeta)$ must be changed. In case $\ell' = \ell$, $\alpha(\zeta) \geq \gamma_1(\zeta)$ (Eq. (155)) to assure that the rays will not emerge through the inside surface and a new constraint on $\omega(\zeta)$ is required to assure that the rays are still going inward when they emerge. In figure 8, it is seen that if a ray has azimuth ϕ at $r = r_i$, then it must go a horizontal distance $d = r_i \cos\phi$ before beginning to go outward. But the horizontal distance that a ray in fan $F_{q'}$ travels in changing its altitude by ζ is $\tau\zeta$, where $\tau = (1 - \mu^2)^{\frac{1}{2}}/\mu$. Therefore,

$$\beta(\zeta) = \arccos\left(\frac{\tau\zeta}{r_i} \wedge 1\right) \tag{158}$$

is the upper limit of the azimuth of rays entering zone $Z_{i,j}$ in fan $F_{q'}$ at a point lying ζ below the top boundary that are still going inward when they reach the top. The $\wedge\, 1$ part of the formula forces $\beta(\zeta) = 0$ when $\tau\zeta \geq r_i$, which is to be desired since all rays entering farther from the top than r_i/ζ are outward-going when they reach the top. Thus, when $\ell' = \ell$,

$$\begin{aligned} \alpha(\zeta) &= (\ell' - 1)\,p \vee \gamma_1(\zeta) \\ \omega(\zeta) &= \ell' p \wedge \beta(\zeta) \end{aligned} \tag{159}$$

When $\ell' = \ell_{max} - \ell + 1$, it is required that the rays emerge going outward through the top. $\alpha(\zeta) \geq \beta(\zeta)$ will assure that they are going outward. A new restriction is required to eliminate rays that leave through the outside boundary. Recalling that $\tau\zeta$ is the horizontal distance traveled in rising to the top, it is seen in figure 9 that

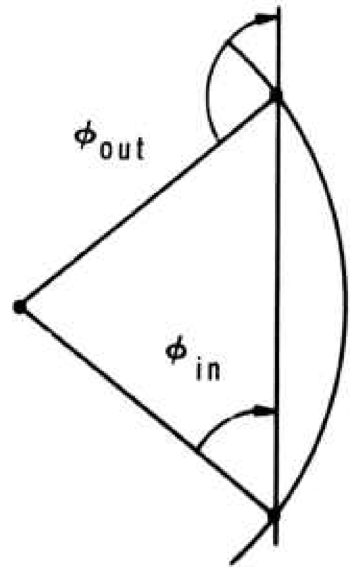


Figure 7. Relation of ϕ_{in} to ϕ_{out}

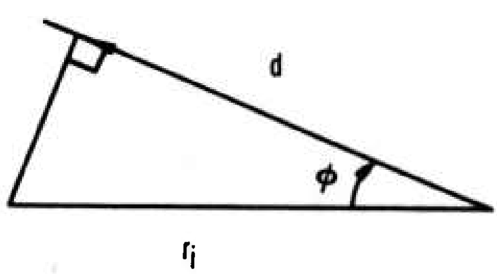


Figure 8. Distance to the Point Closest to the Axis

$$\gamma_2(\zeta) = \arccos\left(\frac{\tau\zeta}{2r_i} \wedge 1\right) \tag{160}$$

is the limiting azimuth. Again, clipping the cosine at 1 is appropriate because any ray that enters the cylinder $r = r_i$ with axial projection μ will leave before it goes an axial distance of $2r_i/\tau$. Thus, when $\ell' = \ell_{max} - \ell + 1$,

$$\begin{aligned} \alpha(\zeta) &= (\ell' - 1)\, p \vee \gamma_1(\zeta) \vee \beta(\zeta) \\ \omega(\zeta) &= \ell'\, p \wedge \gamma_2(\zeta) \end{aligned} \tag{161}$$

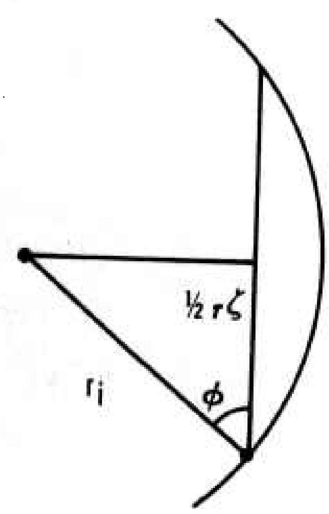


Figure 9. Azimuth of a Ray that Strikes the Upper Outside Corner

<u>Outside-to-Outside</u>. As in the second subcase of outside-to-top, it can be inferred that if $F_{q'}$ consists of rays entering zone $Z_{i,j}$ through the boundary $r = r_i$ and F_q consists of rays leaving through the same boundary, then $\eta_{q,q'} = 0$ unless $\ell' = \ell_{max} - \ell + 1$. The limit on $\alpha(\zeta)$ is simply to assure that the rays emerge through the outside boundary before reaching the top or inside. Thus,

$$\alpha(\zeta) = (\ell' - 1)\, p \vee \gamma_1(\zeta) \vee \gamma_2(\zeta)$$

$$\omega(\zeta) = \ell' \; p. \tag{162}$$

Inside-to-Outside. As is to be expected, this case bears a strong resemblance to the outside-to-inside case. One difference is that the integrals in the numerator and denominator of Eq. (154) are now negative, although $\eta_{q,q'}$ remains positive. Other differences will appear as the discussion progresses.

Let the rays common to fans F_q and $F_{q'}$ enter zone $Z_{i,j}$ through the inside boundary and leave through the outside boundary. As before, the boundary associated with $F_{q'}$ is the one through which the rays are entering; i.e., the inside boundary associated with F_q is the outside boundary. The rays in fan F_q leave the cylinder $r = r_i$ with azimuths in the range $(\ell - 1)\ p \leq \phi \leq \ell p$. It is seen from figure 10 that a ray having azimuth ϕ at $r = r_i$ has azimuth π-arcsin $(r_i \sin \phi)/r_{i-1}$ at $r = r_{i-1}$

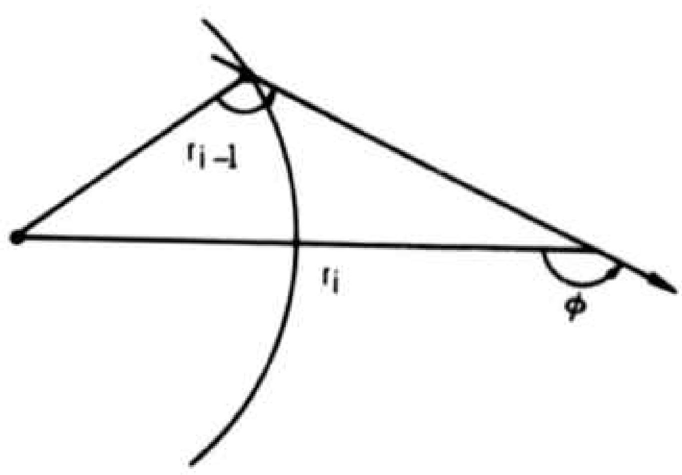


Figure 10. Azimuth of Outgoing Ray

provided that it gets that close to the axis (which is the case if, and only if, $r_i \sin\phi \le r_{i-1}$). Thus, the azimuths of rays of F_q that come within r_{i-1} of the axis lie at that point in the range

$$\pi - \arcsin\left(\frac{r_i \sin(\ell - 1)p}{r_{i-1}} \wedge 1\right) \le \phi \le \pi - \arcsin\left(\frac{r_i \sin \ell p}{r_i} \wedge 1\right)$$

As for the condition that the rays hit the outside boundary before they reach the top, assume that $(r_i - r_{i-1})^2 \le \tau^2 \zeta^2 \le r_i^2 - r_{i-1}^2$. Then a ray of azimuth

$$\phi = \arccos \frac{r_{i-1}^2 + \tau^2 \zeta^2 - r_i^2}{2 r_{i-1} \tau\zeta}$$

will rise by an amount ζ in going from $r = r_{i-1}$ to $r = r_i$. Consequently, if $\gamma_2(\zeta)$ is redefined as

$$\gamma_3(\zeta) = \begin{cases} \pi & \text{if} \quad \tau\zeta \le r_i - r_{i-1} \\ \pi/2 & \text{if} \quad \tau^2\zeta^2 > r_i^2 - r_{i-1}^2 \\ \arccos \dfrac{r_{i-1}^2 + \tau^2\zeta^2 - r_i^2}{2 r_{i-1} \tau\zeta} & \text{otherwise,} \end{cases} \tag{163}$$

then it can be stated that a ray with azimuth ϕ at $r = r_{i-1}$ will reach $r = r_i$ before it rises by an amount ζ if, and only if, $\phi > \gamma_3(\zeta)$. Thus, Eq. (156) is still a valid equation for $\eta_{q,q'}$ provided that

$$\begin{aligned} \alpha(\zeta) &= (\ell' - 1)\, p \vee \left[\pi - \arcsin\left(\frac{r_i \sin(\ell - 1)p}{r_{i-1}} \vee 1\right)\right] \vee \gamma_3(\zeta) \\ \omega(\zeta) &= \ell'\, p \wedge \left[\pi - \arcsin\left(\frac{r_i \sin \ell p}{r_{i-1}} \wedge 1\right)\right] \end{aligned} \tag{164}$$

Inside-to-Top. This case is exactly like the inside-to-outside case except that $\gamma_3(\zeta)$ is now an upper limit:

$$\alpha(\zeta) = (\ell' - 1)\, p \vee \left[\pi - \arcsin\left(\frac{r_i \sin(\ell - 1)\, p}{r_{i-1}} \wedge 1\right)\right]$$

$$\omega(\zeta) = \ell' p \wedge \left[\pi - \arcsin\left(\frac{r_i \sin \ell p}{r_{i-1}} \wedge 1\right)\right] \wedge \gamma_3(\zeta) \tag{165}$$

Next, consider those cases in which $F_{q'}$ is a fan associated with the bottom of zone $Z_{i,j}$ (assuming, as usual, that $\mu > 0$). In such cases the formula (150) for $\eta_{q,q'}$ becomes

$$\eta_{q,q'} = \frac{2\pi\mu}{k_{q'}} \delta_{q'} \int_{r_{i-1}}^{r_i} ([\omega(r) - \alpha(r)] \vee 0)\, r\, dr \tag{166}$$

where $\omega(r)$ and $\alpha(r)$ are the upper and lower limits on the azimuths of rays common to fans F_q and $F_{q'}$ that enter the bottom of $Z_{i,j}$ at radius r.

Bottom-to-Inside. Let the rays common to fans F_q and $F_{q'}$ enter zone $Z_{i,j}$ through the bottom and leave through the inside boundary. The azimuth ϕ of a ray in fan $F_{q'}$ lies in the range $(\ell' - 1)\, p \leq \phi \leq \ell' p$ at $r = r_i$ and hence in the range

$$\arcsin\left(\frac{r_i \sin(\ell' - 1)p}{r} \wedge 1\right) \leq \phi \leq \arcsin\left(\frac{r_i \sin \ell' p}{r} \wedge 1\right)$$

at r. The azimuth ϕ of a ray in fan F_q lies in the range $(\ell - 1)p \leq \phi \leq \ell p$ at $r = r_{i-1}$ and hence in the range

$$\arcsin \frac{r_{i-1} \sin(\ell - 1)p}{r} \leq \phi \leq \arcsin \frac{r_{i-1} \sin \ell p}{r}$$

at r. It remains to determine what limit must be placed on the azimuth of rays entering through the bottom to assure their leaving through the inside boundary. In rising by an amount h, a ray goes a horizontal distance of $h\tau$.

Thus, if $\gamma_4(r)$ is defined as

$$\gamma_4(r) = \begin{cases} 0 & \text{if} \quad r > r_{i-1} + h\tau \\ \arcsin r_{i-1}/r & \text{if} \quad r^2 < r_{i-1}^2 + h^2\tau^2 \\ \arccos \dfrac{r^2 + h^2\tau^2 - r_{i-1}^2}{2\, r\, h\tau} & \text{otherwise} \end{cases} \tag{167}$$

then rays entering the bottom at radius r will reach the inside before the top if, and only if, they enter with azimuth less than $\gamma_4(r)$. Thus, Eq. (166) may be used for $\eta_{q,q'}$, provided that

$$\alpha(r) = \arcsin\left(\frac{(r_i \sin(\ell' - 1)p) \vee (r_{i-1} \sin(\ell - 1)p)}{r} \wedge 1\right)$$
$$\omega(r) = \arcsin\left(\frac{(r_i \sin \ell' p) \vee (r_{i-1} \sin \ell p)}{r} \wedge 1\right) \wedge \gamma_4(r) \tag{168}$$

<u>Bottom-to-Top</u>. Let $F_{q'}$ still be associated with the bottom of zone $Z_{i,j}$, but let F_q be associated with the top. As in the outside-to-top case, $\eta_{q,q'} = 0$ unless $\ell' = \ell$ or $\ell + \ell' = \ell_{max} + 1$, but now $\ell' = \ell$ splits into two subcases: $\ell' = \ell \leq (1/2)\ell_{max}$ and $\ell' = \ell > (1/2)\ell_{max}$. Also, it will be necessary to redefine β and γ. Let

$$\beta_1(r) = \arccos\left(\frac{\tau h}{r} \wedge 1\right) \tag{169}$$

so that rays having azimuth $\phi < \beta_1(r)$ at r will still be going inward after rising by an amount h. And let

$$\gamma_5(r) = \begin{cases} 0 & \text{if} \quad r < h\tau - r_i \\ \pi & \text{if} \quad r < r_i - h\tau \\ \arccos \dfrac{r^2 + h^2\tau^2 - r_i^2}{2\, r\, h\tau} & \text{otherwise} \end{cases} \tag{170}$$

so that rays having azimuth $\phi < \gamma_5(r)$ at r will rise by an amount h before emerging from the cylinder $r = r_i$.

Consider first the subcase $\ell' = \ell \leq (1/2)\ell_{max}$, which is the one where the rays are still going inward as they leave through the top of the zone. Then

$$\begin{aligned} \alpha(r) &= \arcsin\left(\frac{r_i \sin(\ell' - 1)p}{r} \wedge 1\right) \vee \gamma_4(r) \\ \omega(r) &= \arcsin\left(\frac{r_i \sin \ell' p}{r} \wedge 1\right) \wedge \beta_1(r) \end{aligned} \tag{171}$$

Next, if $\ell' \leq (1/2)\ell_{max}$ and $\ell = \ell_{max} - \ell' + 1$ ($> (1/2)\ell_{max}$), then

$$\begin{aligned} \alpha(r) &= \arcsin\left(\frac{r_i \sin(\ell' - 1)p}{r} \wedge 1\right) \vee \beta_1(r) \vee \gamma_4(r) \\ \omega(r) &= \arcsin\left(\frac{r_i \sin \ell' p}{r} \wedge 1\right) \wedge \gamma_5(r) \end{aligned} \tag{172}$$

and, finally, if $\ell' = \ell > (1/2)\ell_{max}$, then

$$\begin{aligned} \alpha(r) &= \pi - \arcsin\left(\frac{r_i \sin(\ell - 1)p}{r} \wedge 1\right) \\ \omega(r) &= \left[\pi - \arcsin\left(\frac{r_i \sin \ell' p}{r} \wedge 1\right)\right] \wedge \gamma_5(r) \end{aligned} \tag{173}$$

<u>Bottom-to-Outside</u>. Let $F_{q'}$ still be a fan associated with the bottom of zone $Z_{i,j}$, but let F_q be associated with the outside boundary. Then $\eta_{q,q'} = 0$ unless $\ell' = \ell$ or $\ell' = \ell_{max} - \quad + 1$. Since the rays of F_q are assumed to be leaving $Z_{i,j}$ at the outside boundary, $\ell > (1/2)\ell_{max}$. Thus, for $\ell' = \ell_{max} - \ell + 1$,

$$\alpha(r) = \arcsin\left(\frac{r_i \sin(\ell' - 1)p}{r} \wedge 1\right) \vee \gamma_5(r)$$
$$\omega(r) = \arcsin\left(\frac{r_i \sin \ell' p}{r} \wedge 1\right) \qquad (174)$$

and for $\ell' = \ell$,

$$\alpha(r) = \left[\pi - \arcsin\left(\frac{r_i \sin(\ell' - 1)p}{r} \wedge 1\right)\right] \vee \gamma_5(r)$$
$$\omega(r) = \pi - \arcsin\left(\frac{r_i \sin \ell' p}{r} \wedge 1\right) \qquad (175)$$

ESTIMATION OF THE LENGTH

The factor t_q occurring in Eq. (146) is a quantity associated with fan F_q having the dimensions of length. It represents something like the mean length of the intersections of rays of fan F_q with the zone it leaves. The procedure used in SUBROUTINE DRAW of SHORT2 is to choose a representative ray of fan F_q and set t_q equal to the length of its intersection with the zone. In order to specify precisely what is done, define the mean direction of cone $C_{\ell,m}$ to be one with axial projection $\mu = \mu_{m-\frac{1}{2}}$ and azimuth $\phi = (\phi_{\ell-1,m} + \phi_{\ell,m})/2$ and the mid-circle of boundary $BA_{i,j}$ to be the circle $z = z_j$, $r = (1/2)(r_{i-1} + r_i)$ and the mid-circle of boundary $BR_{i,j}$ to be the circle $z = (z_{j-1} + z_j)/2$, $r = r_i$. In these terms, if F_q is associated with boundary B_q and cone C_q, then a representative ray R_q of F_q is a ray passing through the mid-circle of B_q in the mean direction

of cone C_q, and t_q is the length of the intersection of R_q with Z_q^o. The value of t_q is well-defined, because all representative rays of F_q are equivalent with respect to the symmetry of the system, and consequently they intersect Z_q^o in segments of equal length.

THE SOURCE TERM

The strength of the source at the center of a HECTIC zone is calculated from zone-centered temperatures using Eq. (131), and at other points it is obtained by a linear (in r and z) interpolation of the zone-centered data. The source term S_q in Eq. (146) is then obtained in the following way. Let R_q be the representative ray of F_q as defined above, and let P^i and P^o, respectively, be the points where R_q enters and leaves zone Z_q^o. The distance between these points has already been given the designation t_q. Let S^α be the strength of the source at P^α, α = i or o. The contribution to the intensity of radiation at points along the segment of R_q between P^i and P^o due to the source in zone Z_q^o is assumed to satisfy the transport equation,

$$\frac{dI}{ds} = - \sigma I + S^i + \frac{s(S^o - S^i)}{t_q}$$

where s is the distance along R_q from P^i toward P^o and $I(0) = 0$. It follows that

$$I(t_q) = \left[S^i - \frac{S^o - S^i}{\sigma t_q}\right] \frac{1 - e^{-\sigma t_q}}{\sigma} + \frac{S^o - S^i}{\sigma} \tag{176}$$

Finally, the source term is defined to be

$$S_q = I(t_q)\, k_q \tag{177}$$

where k_q is the number of rays in fan F_q as given in Eq. (147).

Another type of source to be considered is the radiant energy that flows into the system through its outer surface. As in LONG2, a boundary temperature θ_{out} is assigned to each interval of the outer boundary by input cards. On the assumption that the incoming energy is isotropic black-body radiation, the energy flow assigned to an incoming fan F_q is

$$J_q = \frac{ac}{4\pi} \theta_{out}^4 \left[P\left(\frac{h\nu_2}{\theta_{out}}\right) - P\left(\frac{h\nu_1}{\theta_{out}}\right) \right] k_q \tag{178}$$

which is the SHORT2 version of Eqs. (133) and (134). To see the analogy fully, observe that for any outer boundary interval B,

$$\sum_{q \in Q} k_q = \pi\alpha \tag{179}$$

where α is the area of B and $\{F_q\}_{q\in Q}$ is the set of fans F_q for which B_q = B and which enter the system through B.

Serious difficulties have been encountered when the treatment of sources just described has been applied to overly heterogeneous systems. If, for example, a hot opaque zone lies next to a cold transparent one, then representative rays running through the cold zone after emerging from the hot one will, according to Eq. (176) pick up far more intensity than they should because of the large value of S^i. In LONG2, the problem has been relieved to some extent by setting

$$I(t^q) = S^c \frac{1 - e^{-\sigma t_q}}{\sigma} \tag{180}$$

where S^c is the central source strength whenever $S^i \wedge S^o \vee S^c < f(S^i \vee S^o \vee S^c)$, where f is input (usually, f = 2).

It must be emphasized that the problem of interpolating a source distribution on the basis of data supplied on a coarse mesh will never be completely solved. But in view of the expense of performing two-dimensional transport calculations, there is a strong economic incentive to find partial solutions of various types.

SOME SPECIAL SOLUTIONS

Two cases of importance are easy to solve with the view-factor method. The first is the case of an evacuated system with uniform isotropic incident radiation. In this case, $S = \sigma = 0$ in all zones, and Eq. (176) becomes indeterminate; but the correct assignment is, of course, $S_q = 0$. Let the energy flow in the incoming fans F_q be

$$J_q = I^o k_q$$

By virtue of the relation expressed in Eq. (136), $J_q = I^o k_q$ for all q is the solution to Eq. (146).

Next, suppose that the system is filled with material of constant absorption coefficient and constant source strength $S = \sigma I^o$. If, in addition, radiant energy is flowing into the system with isotropic intensity I^o, then $J_q = I^o k_q$ again satisfies Eqs. (146), (176), and (177) with $I(t_q) = I^o (1 - e^{-\sigma t_q})$.

Another important special case, but one for which there exist no trivial solutions, is the case of the thick zone. To say a zone is thick implies that $\sigma t_q >> 1$ and

$$I(t_q) \simeq \frac{S^o}{\sigma} - \frac{S^o - S^i}{\sigma^2 t_q}$$

Suppose, for example, that $S = S_o + z S_1$ so that $S^o - S^i = \Delta z S_i$ and $t_q = \Delta z/\mu_q$, where Δz is the difference in altitude between P^o and P^i. Thus,

$$I(t_q) \simeq \frac{S^o}{\sigma} - \frac{\mu_q S_1}{\sigma}$$

and

$$J_q \simeq \left(\frac{S^o}{\sigma} - \frac{\mu S_1}{\sigma^2}\right) k_q$$

Next, let B be some horizontal zone boundary, and let Q be the set of fan indices q for which $\mu_q > 0$ and $B_q = B$. Then the rate of transfer of radiant energy in the upward direction through B will be, according to the view-factor method,

$$\dot{E}_B^{up} = \sum_{q \in Q} J_q$$

$$= \frac{S^o}{\sigma} \sum_Q k_q - \frac{S_1}{\sigma^2} \sum_Q \mu_q k_q$$

To perform the sums, first sum over azimuth. Simply setting $\phi_\ell = \pi/2$ and $\phi_{\ell-1} = 0$ in Eq. (152) gives 1/4 of the total, i.e.,

$$k_q = \frac{\pi \delta_q}{2} (\pi r_i^2 - \pi r_{i-1}^2) \mu_q$$

so the sum over azimuth is

$$\sum_\ell k_q = 2 \pi \delta_q \alpha \mu_q$$

where $\alpha = \pi(r_i^2 - r_{i-1}^2)$, the area of B. The sums over axial projection are then

$$\sum_m \mu_{m-\frac{1}{2}} (\mu_{m-1} - \mu_m) \approx \int_0^1 \mu d\mu = \frac{1}{2}$$

$$\sum_m \mu^2_{m-\frac{1}{2}} (\mu_{m-1} - \mu_m) \approx \int_0^1 \mu^2 d\mu = \frac{1}{3}$$

which finally leads to

$$\dot{E}_B^{up} \simeq \frac{S^o}{\sigma} \pi\alpha - \frac{S_1}{2} \frac{2\pi\alpha}{3}$$

Performing the same calculation for $\mu < 0$ only changes the sign of the first term. Thus,

$$\dot{E}_B = \dot{E}_B^{up} + \dot{E}_B^{down} = -\frac{4\pi\alpha}{3}\frac{S_1}{\sigma^2}$$

or, since

$$\frac{S_1}{\sigma} = \frac{1}{\sigma}\frac{dS}{dz} = \frac{dS}{d\tau}$$

where τ is optical depth, one finally arrives at the diffusion approximation

$$\dot{E}_B = -\frac{4\pi\alpha}{3\sigma}\frac{dS}{d\tau}$$

A similar result is obtained when the gradient of the source is radial, but there are additional complications in that case caused by the curvature of vertical boundaries.

METHOD OF SOLUTION

Solving the transport equation in the view-factor approximation is, on the face of it, a matter of solving a large linear system of simultaneous equations (Eq. (146)). It turns out, however, that it is possible to arrange the equations in a sequence that permits each to be solved in terms of the solution of the preceding ones.

Recall first that $\eta_{q,q'} = 0$ unless $m_q = m_{q'}$. Hence, it is possible to solve subsystems of equations corresponding to the various values of m_q independently of each other. Consider, then, the problem of solving the system represented by Eq. (146) for all q such that $m_q = m$. Suppose, for example, that the sign of $\mu_{m-\frac{1}{2}}$ is positive. The technique is to treat a layer of zones at a time, a layer of zones being $Z_{i,j}$ for $1 \leq i \leq i_{max}$ and j fixed, starting with the bottom layer ($j = 1$) and proceeding upward. Within a layer, one first calculates the J's of the incoming fans ($\phi_\ell \leq \pi/2$), starting at the outside and proceeding inward, and then the outgoing fan ($\phi_\ell > \pi/2$), starting at the center ($i = 1$) and proceeding outward. To see why this

sequence has all the desired properties, observe first that when $j = 1$, all the J's for fans entering through the bottom are specified as boundary sources*, which is also true of incoming fans at $i = i_{max}$. On the inward sweep of the bottom layer, the J's of incoming fans entering through the bottom and outside of any particular zone $Z_{i,1}$ are all that are needed to calculate the incoming fans leaving through the inside and top. Since the incoming fans leaving $Z_{i,1}$ through $BL_{i,1}$ are the incoming fans entering $Z_{i-1,1}$ through $BR_{i-1,1}$ (= $BL_{i,1}$), the next step is ready to be taken, and the validity of the inward sweep is established. To calculate J's of the outgoing fans leaving $Z_{i,j}$ through the outside and top, it suffices to have the J's of the outgoing fans entering through the inside and bottom together with the incoming fans that were obtained on the inward sweep. Thus, an outward sweep does permit the calculation of the outgoing fans. The process produces the J's of both incoming and outgoing fans through the top boundaries of the layer which provides the bottom boundary condition for the layer above. Thus, the entire upward sweep is carried out. When the upward sweep is done, a downward sweep is performed with the same set of η's and t's, but with the layers reversed for the μ which is just the negative of the one involved in the preceeding upward sweep.

The contributions to $\dot{E}_{i,j}$ (Eq. (138)) are accumulated as the J's are computed. Some storage of J's is required, but not much; it suffices to have room for all the J's that enter or leave one layer for one value of μ. When the top boundary is a symmetry boundary, the appropriate boundary condition is $J_{\ell,m'} = J_{\ell,m}$, where $\mu_{m'} = -\mu_m$. When the bottom boundary is a symmetry boundary, a similar relationship subsists on the bottom, so the downward sweep must precede the upward one. A pair of symmetry planes, which would imply an axial periodicity, is not currently treated by the code.

* As discussed in the section entitled "The Source Term."

SECTION VI

FLOW OF CONTROL IN SUBROUTINE DRAW AND THE "DRAW SUBROUTINE SEQUENCE"

SUBROUTINE DRAW specifies the overall flow of control through the "DRAW subroutine sequence" (shown in Figs. 11 and 12) that initializes geometric quantities and creates the "DRAW data" for the "transport subroutine sequence" (shown in Fig. 18).

Upon entering SUBROUTINE DRAW the input to the "DRAW subroutine sequence" is read in on cards. This input consists of black-body system boundary temperatures, two tolerances that place restrictions on accuracy of the calculations in the "DRAW subroutine sequence" (TOLER and CHKSUM), and variables LMAX and MUMAX that specify the number of fans of direction to be constructed per hydro zone of a HECTIC problem.

The "DRAW subroutine sequence" creates, for each zone surface of the hydro mesh, a set of fans of direction, each specified by a polar angle θ_m(arccos MU(M)), an azimuthal angle Φ_ℓ(PHI(L)), and a range $\Delta\Phi$(PP) of azimuthal angles that must be the same for all fans associated with all zone surfaces. Each zone surface has associated with it a set of fans, each set consisting of LMAX*MUMAX*2 fans. (For each polar angle θ_m of MUMAX polar angles, there are LMAX*2 azimuthal fans.) Because of cylindrical symmetry, however, only LMAX azimuthal fans are considered in the "DRAW subroutine sequence."

Given a set of fans for each zone face, the "DRAW subroutine sequence" performs a sweeping operation through the hydro mesh in a specified manner calculating "view factors," or "η's," which couple

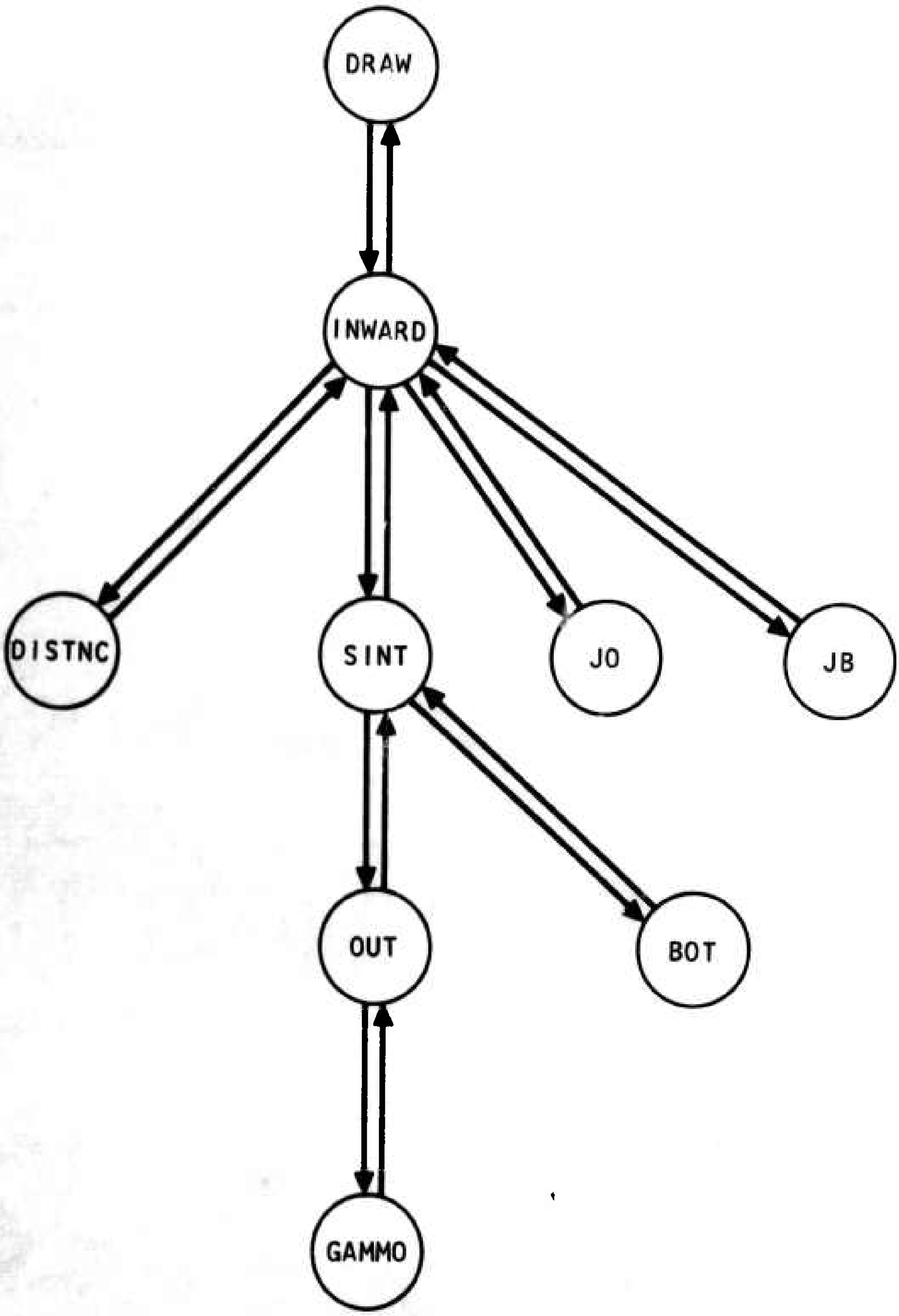


Figure 11. SUBROUTINES of the "DRAW Subroutine Sequence" Employed in an Inward DRAW Sweep

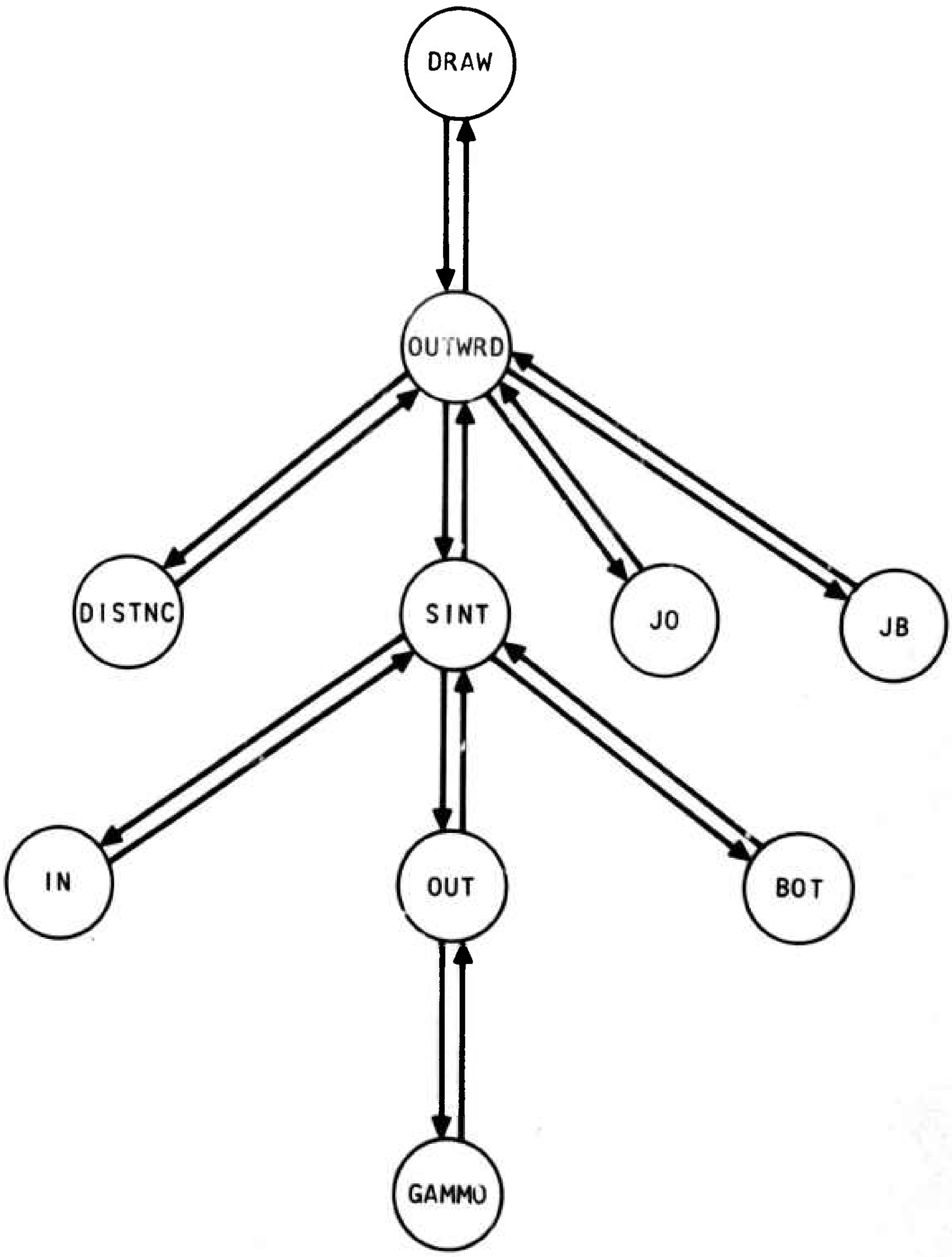


Figure 12. SUBROUTINES of the "DRAW Subroutine Sequence" Employed in an Outward DRAW Sweep

each fan on each zone face with all the other fans on all the neighboring zone faces. Formally:

Given a particular hydro zone,

$\eta_{S_1 \to S_2}(M, L, LP)$ = the fraction of (energy entering the zone in time t through face S_1 in fan LP) which leaves the zone through face S_2 in fan L in the case of isotropic, homogeneous radiation $\sigma = S = 0$ with the intensity of the radiation field, I_o, = 1.

Notice the neighboring zone faces S_2, are the faces of the zones which share face S_1.

The order in which sweeping occurs in DRAW is as follows:

For each polar angle specified in M (M=1,MUMAX/2), sweeping is performed upward through the hydro mesh. For each zone horizontal layer, an inward and or outward sweep is performed. On the inward sweep, sweeping begins at the system outside boundary. (For an account of which η's are calculated on the inward and outward sweeps, see SUBROUTINES INWARD and OUTWARD respectively). Only half the polar angles are considered, because

$$\eta_{S_1 \to S_2}(M, L, LP) = \eta_{S_1 \to S_2}(MUMAX+1-M, L, LP)$$

due to the symmetry restrictions placed on the polar angles θ_M (see DRAW Glossary).

Notice that a fan of a given zone face may be associated with any point on that face. For this reason each η calculation consists of an integration over the given zone face, and this integration is performed numerically by FUNCTION SINT under control of the relative error criterion TOLER, which is input in DRAW. It is desirable and reasonable to require that the η's be calculated such that $\Sigma\eta(L, LP) = 1.0$ exactly, for each fan LP of a zone face summed over all fans L of neighboring zone faces. However, each η is calculated by an integration

with a certain error; and with the present version of the code the η's do not sum exactly to 1.0. The sum is checked in DRAW (and partially in SUBROUTINE OUTWRD) for each fan of each zone face and a message is printed if the sums are not within CHKSUM (input) of 1.0. If the printing of this message occurs during execution of SUBROUTINE DRAW, it is an indication (assuming CHKSUM $\gtrsim$ 10* TOLER) that there is a programming error in the calculation of the η's. The η's along with information specifying the characteristic ray path (see SUBROUTINE DISTNC) associated with each fan on each zone face and other information comprise the "DRAW data" which is output by SUBROUTINES DRAW, INWARD, and OUTWRD to a "DRAW data" file to be used by SUBROUTINE TRAN2 in each cycle of a HECTIC problem. Output of this data is accomplished by the filling and subsequent emptying of an output storage buffer (BUFF2) as the sweeping through the hydro mesh occurs, creating records of data on the file as the buffer is filled and emptied. The smallest unit of data allowed for one buffer-load is an inward or an outward sweep through one zone (one CALL executed to SUBROUTINES INWARD or OUTWRD). The end of an inward or outward sweep of one zone layer is always terminated by the creation of a record on the "DRAW data" file; thus, "DRAW data" of one complete sweep of a zone layer is never mixed with that of the following or preceding zone layer sweep (this ensures ease of handling this "DRAW data" in SUBROUTINE TRAN2).

It is intended that the "DRAW data" file be a magnetic tape. Since the data comprising this file is of a geometric nature and only depends on the HECTIC problem mesh and the variables LMAX, MUMAX, and TOLER, a complete HECTIC problem (comprising many cycles) may be solved with only one FORTRAN CALL to SUBROUTINE DRAW and subsequent CALLS to SUBROUTINE TRAN2 for each cycle of a HECTIC problem. If the "DRAW data" file is a magnetic tape, then SUBROUTINE DRAW need not be called again to regenerate data. (However, parameters transferred to TRAN2 by DRAW through COMMON must be initialized.)

Input to "DRAW SUBROUTINE SEQUENCE"

Medium	Variables
CARDS	LMAX
	MUMAX
	TOLER
	CHKSUM
	ATHETA(I), I=1, IMAX
	BTHETA(I), I=1, IMAX
	RTHETA(J), J=1, JMAX

Conditions That Cause Termination of Execution in "DRAW SUBROUTINE SEQUENCE

The current version of the coding has only one error condition test. BUFF1 in SUBROUTINES INWARD and OUTWRD is not allowed to overflow--the problem will stop if this occurs (see SUBROUTINES INWARD or OUTWRD). There are, however, restrictions on the variables that are considered input to the "DRAW subroutine sequence" and restrictions on variables that are to be calculated by the sequence. (See SUBROUTINE DRAW Glossary.)

Results from DRAW SUBROUTINE SEQUENCE Transferred to the "TRANSPORT SUBROUTINE SEQUENCE"

Medium	Variables
COMMON	LMAX
	MUMAX
	ATHETA(I), I=1, IMAX
	BTHETA(I), I=1, IMAX
	RTHETA(J), J=1, JMAX

	PP
	L2
	L2P1
	PHI(L), L=1, LMAX
	COSPHI(L), L=1, LMAX
	SINLP(L), L=1, LMAX
	SINP2
	M2
	MU(M), M=1, MUMAX
	SMU(M), M=1, MUMAX
	W(M), M=1, MUMAX
	NOBUFF
"DRAW data" FILE	NOO
	CH, CVIN, CVOUT
	DH, XH, YH
	DV, XV, YV
	All η's

<u>Glossary of Variables used in SUBROUTINE DRAW</u>

FORTRAN Label	Report Label	Description
ATHETA(I)	θ_{out}	The temperature of the black-body system boundary adjacent to zone (I, JMAX) (in eV) (input on cards) (used in TRAN2)
BUFF2	None	≡ P. BUFF2(1) = NOO. Output storage buffer array for the "DRAW data" that is to be written on the "DRAW data" file. (The maximum number of words allowed is 1023, and each buffer-load must contain at least one complete (inward or outward) zone sweep.)
CAP	None	Saved temporarily on disk in DRAW

CHKSUM	None	Error criterion for the sum of the η's: For each polar direction specified by M, each hydro zone (I,J), each face S_1 of these hydro zones, and each fan LP on faces S_1 it is desirable, letting $S = \sum_{\substack{\text{neighboring} \\ \text{zone faces } S_2 \\ \text{and all fans} \\ \text{L on each of} \\ \text{these faces.}}} \eta_{S_1 \to S_2}$ (M, L, LP) (See SUMB, SUMO) that $\lvert S-1 \rvert$ be < CHKSUM. All failures to meet this criterion will result in a printed message (input on card) (should be > TOLER)
COSPHI(L)	$\cos \Phi_\ell$ $= [\cos (\ell-\frac{1}{2})\Delta\Phi]$	Cos(PHI(L))
FIOUT	None	Saved temporarily on disk in DRAW
I	i	Index of the current radial hydro zone with outer radius X(I)
IBB	None	An indicator: IBB = 0 means for the current M and J, the "bottom η's" sum to 1 IBB = 1 means for the current M and J, the "bottom η's" do not sum to 1 (See SUMB)
ICHECK	None	An indicator: ICHECK = 0 means for the current M and J, the "inside η's" sum to 1. ICHECK = 1 means for the current M and J, the "inside η's" do not sum to 1. (See SUM in OUTWRD Glossary.)
IMAX	i_{max}	The number of radial hydro zones

IOO	None	An indicator: IOO = 0 means for the current M and J, the "outside η's" sum to 1. IOO = 1 means for the current M and J, the "outside η's" do not sum to 1. (See SUMO)
J	j	Index of the current horizontal layer of hydro zones with upper boundary Y(J)
JMAX	j_{max}	The number of vertical hydro zones
K	None	A running index
KK2	None	The index of the location in the BUFF2 array which contains the last entry of "DRAW data" for the current filling of the array
KMAX	None	KMAX-1 is the total number of hydro zones in the system
KMAXA	None	KMAX+1
L	ℓ	A running index
L2	None	LMAX/2
L2P1	None	L2+1
LMAX	ℓ_{max}	The total number of fans of directions to be considered in the "DRAW subroutine sequence." (input on card) (must be even and ≤ 6)
M	m	Index of the current polar angle θ_m (arccos MU(M)) being considered in constructing the η's
M2	None	MUMAX/2

MU(M)	μ_m	Cos θ_m, where θ_m is the m^{th} discrete polar direction (along which the η's are calculated) measured in a clockwise sense from the vertical. $(\theta_m = \frac{\pi(m-\frac{1}{2})}{MUMAX})$; $0 < \theta_m < \pi$; $\Delta\theta = \theta_{m+\frac{1}{2}} - \theta_{m-\frac{1}{2}}$ is constant for all m; θ_m, m=1, MUMAX symmetric about $\pi/2$.)
MUMAX	2M	The maximum number of discrete polar directions, θ_m, considered in the construction of the η's. (input on card) (must be even and ≤ 6)
N	None	A running index
NOBUFF	None	The number of output buffer (BUFF2) loads filled per inward and outward sweep of DRAW along one horizontal layer of zones (J) in direction θ_m. (Not a function of M or J)
NOO	None	BUFF2(1) The number of inward and/or outward zone sweeps contained in the current output buffer load BUFF2. (Each buffer load must contain at a minimum one complete (inward or outward) zone sweep)
P	None	Saved temporarily on disk in DRAW
PHI(L)	Φ_ℓ $= (\ell-\frac{1}{2})\Delta\Phi$	The average azimuthal direction of fan L measured in the horizontal plane containing the vector whose origin is the point in question and terminal point is the axis, in a positive sense away from the vertical axis, $(0 < \Phi_\ell < \pi; \Phi_\ell, \ell=1$, LMAX symmetric about $\pi/2$; $\Delta\Phi = \Phi_\ell - \Phi_{\ell-1}$ is constant for all ℓ).

PI	π	π
PP	$P = \Delta\Phi = \Phi_{\ell+\frac{1}{2}} - \Phi_{\ell-\frac{1}{2}}$	π/LMAX, a constant
RTHETA(J)	θ_{out}	The temperature of the black-body system boundary adjacent to zone (IMAX, J) (in eV) (input on cards) (used in TRAN2)
SINLP	$\sin(\ell \cdot \Delta\Phi)$	Sin (PP*L)
SINP2	None	Sin $(\frac{PP}{2})$
SMU(M)	$\sqrt{1-\mu_m^2}$	$\mathrm{Sin}(\theta_m)$
STEF	$\frac{ac}{4}$	Stefan's constant = 1.0283×10^{12} (in $\frac{ergs}{cm^2 \sec eV^4}$)
SUMB(I, LP)	None	Given directions specified by M and LP and hydro zone (I, J): $\mathrm{SUMB(I,LP)} = \sum_{L=1,L2} \mathrm{ETABI(L,LP)} + \sum_{L=1,LMAX} \mathrm{ETABT(L,LP)} + \sum_{L=L2P1,LMAX} \mathrm{ETABO(L,LP)}$ (See CHKSUM) (should = 1.)
SUMO(I, LP)	None	Given directions specified by M and LP and hydro zone (I, J): $\mathrm{SUMO(I,LP)} = \sum_{L=1,L2} \mathrm{ETAOI(L,LP)} + \sum_{L=1,LMAX} \mathrm{ETAOT(L,LP)} + \sum_{L=L2P1,LMAX} \mathrm{ETAOO(L,LP)}$ (See CHKSUM) (should = 1.)

TOLER	None	Relative error criterion imposed upon each integration performed by SUBROUTINE SINT (in decimal fraction) (input on card)
U	None	Saved temporarily on disk in DRAW
V	None	Saved temporarily on disk in DRAW
W(M)	$\frac{\lvert\Delta\mu_m\rvert}{2} = \frac{\lvert\mu_{m+\frac{1}{2}} - \mu_{m-\frac{1}{2}}\rvert}{2}$	$\lvert\mathrm{Cos}[M*\Delta\theta] - \cos[(M-1)*\Delta\theta]\rvert * \frac{1}{2}$

FLOW OF CONTROL IN SUBROUTINE INWARD

SUBROUTINE INWARD is called by SUBROUTINE DRAW to compute, given polar angle θ_M, the η's that couple for face S_1 of zone (I,J) all the fans LP on S_1 containing rays entering the zone traveling in an inward direction with fans L (on all the other zone faces) which contain rays leaving zone (I,J) traveling in an inward direction. A ray traveling through the system is considered traveling in an inward direction at point P if point P comes before the point D of closest approach between the ray and the system axis; the ray is considered traveling in an outward direction at point P if P comes after D.

Given below are the η's that meet the above restriction for the θ_m's (all $< \pi/2$) considered in the "DRAW subroutine sequence," and which are calculated in this routine:

$\eta_{B\to I}$ (L, LP) L=1, L2; LP=1, L2

Considering rays entering the zone bottom face traveling inward and exiting the zone inside face (rays leaving the zone through the inside face must be traveling inward).

$\eta_{O \to I}$ (L, LP) L=1, L2; LP=1, L2

Considering rays entering the zone outside face (must be traveling inward) and exiting the zone inside face.

$\eta_{B \to T}$ (L, LP) L=1, L2; LP=1, L2

Considering rays entering the zone bottom face traveling inward and exiting the zone top face traveling inward.

$\eta_{O \to T}$ (L, LP) L=1, L2; LP=1, L2

Considering rays entering the zone outside face and exiting the zone top face traveling inward

The computed η's along with other "DRAW data" (see glossary) are stored in storage buffer array BUFF1 as they are calculated. At the end of the inward sweep of zone (I, J), BUFF1 is emptied into the output storage buffer array BUFF2 to be saved for output to the "DRAW data" file. If BUFF2 cannot accommodate the BUFF1 load of new data, it is output to the "DRAW data" file. BUFF1 must be able to store all the "DRAW data" for the inward sweep through zone (I, J); if it cannot, the problem is stopped.

Before control is returned to SUBROUTINE DRAW, preliminary calculations are performed for the sum-check on the η's (see SUBROUTINE DRAW).

Glossary of Variables Used in SUBROUTINE INWARD

FORTRAN Label	Report Label	Description
AA(L)	a_ℓ	At a given radius X(I): AA(L) = SINLP(L)*X(I), the distance of closest approach (to the vertical axis) of a horizontal ray making azimuthal angle (π-L*PP) with the radius vector at X(I) (in cm)

AID	None	Temporary storage
ALPHAL	None	In computing ETABI(L, LP), for a given L and radius X(I): π - (the angle the radius vector makes with ray $\vec{R}$ at X(I)), where ray $\vec{R}$ = a ray traveling inward towards the vertical axis in a horizontal plane making an angle (π - L*PP) with a radius vector at X(I-1) (in radians)
ANGL1	None	Temporary storage for information transferred to SUBROUTINES BOT and OUT
ANGL2	None	Same as above
ASN	None	Given I, ASN = $\arcsin \frac{X(I-1)}{X(I)}$ (in radians)
BUFF1	None	$\equiv$ CAP Storage buffer array for each (inward or outward) zone sweep. When each sweep is completed, the contents of BUFF1 are added to the output storage buffer array BUFF2
BUFF2	None	$\equiv$ P. BUFF2(1) $\equiv$ NOO Output storage buffer array (see DRAW Glossary)
CAP	None	Saved temporarily on disk in DRAW
CH(LP)	k_q	= JB(J-1, I, LP, M) (in ergs/sec) (part of "DRAW data" transferred to TRAN2)
CV(LP)	k_q	= JO(J, I, LP, M) (in ergs/sec) (part of "DRAW data" transferred to TRAN2)

CVIN(L)	k_q	= JO(J,I-1,L,M) (in ergs/sec) (part of "DRAW data" transferred to TRAN2)
DBGPRT	None	The debug print control: DBGPRT $\geq 10^{-20}$ means debug print desired; otherwise, debug print not desired
DELTY	h	Given J, DY(J) (in cm)
DH(L)	t	Given M,J,I,L: The length of the characteristic ray of fan L measured on the top face Y(J) of zone (I,J) (See SUBROUTINE DISTNC) (in cm) (part of "DRAW data" transferred to TRAN2)
DV(L)	t	Given M,J,I,L: The length of the characteristic ray of fan L measured on the vertical face X(I-1) of zone (I,J) (See SUBROUTINE DISTNC) (in cm) (part of "DRAW data" transferred to TRAN2)
DY(J)	$h = z_j - z_{j-1}$	Vertical length of hydro zones (I,J), I=1,IMAX (in cm)
EPS	None	Absolute error criterion imposed upon each integration performed by SUBROUTINE SINT

ETABI(L, LP)	η	Given M, J, I, L, LP: The fraction of (energy in fan LP passing into zone (I, J) through its bottom face in 1 sec) that passes out zone (I, J) through its inside face in fan L, in direction MU(M) in the case of isotropic, homogeneous radiation σ =S=0 (part of "DRAW data" transferred to TRAN2)
ETABT(L, LP)	η	Given M, J, I, L, LP: The fraction of (energy in fan LP passing into zone (I, J) through its bottom face in 1 sec.) that passes out zone (I, J) through its top face in fan L, in direction MU(M) in the case of isotropic, homogeneous radiation σ =S=0 (part of "DRAW data" transferred to TRAN2)
ETAOI(L, LP)	η	Given M, J, I, L, LP: The fraction of (energy in fan LP passing into zone (I,J) through its outside face in 1 sec) that passes out zone (I, J) through its inside face in fan L, in direction MU(M) in the case of isotropic, homogeneous radiation σ =S=0 (part of "DRAW data" transferred to TRAN2)
ETAOT(L, LP)	η	Given M, J, I, L, LP: The fraction of (energy in fan LP passing into zone (I, J) through its outside face in 1 sec) that passes out zone (I, J) through its top face in fan L, in direction MU(M) in the case of isotropic, homogeneous radiation σ =S=0 (part of "DRAW data" transferred to TRAN2)
FORPI	4π	4π
HELP1	None	Temporary storage for information transferred to SUBROUTINE BOT.

HELP2	None	Same as above
I	i	Index of the current radial hydro zone with outer radius X(I)
IAID	None	Flag generated for SUBROUTINES BOT and OUT
ISEND	None	Flag generated for SUBROUTINES BOT and OUT
ISWEEP	None	Current direction of DRAW sweep: = 1 means sweeping inward = 2 means sweeping outward
J	j	Axial index of the current hydro zone with upper boundary Y(J)
KK1	None	Index of the last current entry in the storage buffer array BUFF1 (must not exceed 1023)
KK2	None	Index of the last current entry in the output storage buffer array BUFF2
KKK	None	A running index
L	ℓ	Fan index
L2	None	LMAX/2
L2P1	None	L2+1
LHI	None	Complement of the lowest index of the fans for which characteristic ray lengths are to be calculated in SUBROUTINE DISTNC
L1M1	None	The lower limit in some integrations by SUBROUTINE SINT (in cm)
L1M2	None	The upper limit in some integrations by SUBROUTINE SINT (in cm)
LLOW	None	Complement of the highest index of the fans for which characteristic ray lengths are to be calculated in SUBROUTINE DISTNC
LMAX	ℓ_{max}	The total number of fans of directions to be considered in the "DRAW subroutine sequence" (input on card in DRAW)

LMDA	$h\tau$	Given M and J: The horizontal distance a ray will travel in horizontal zone layer DY(J) between intersections with surface Y(J-1) and Y(J) in direction θ_m; or DY(J)*TAN(θ_m) (in cm)
LP	ℓ'	Fan index
M	m	Index of the current polar angle θ_m(arccos MU(M)) being considered in constructing the η's
MU(M)	μ_m	cos θ_m. (See DRAW glossary)
NOBUFF	None	The number of output buffer (BUFF2) loads filled per inward and outward sweep of DRAW along one horizontal layer of zones(J) in direction μ_m. (Not a function of M or J)
NOO	None	BUFF2(1) The number of inward and/or outward zone sweeps contained in the current output buffer load BUFF2. (Each buffer load must contain at a minimum one complete (inward or outward) zone sweep) (part of "DRAW data" transferred to TRAN2)
PP	$\Delta\Phi = p = \Phi_{\ell+\frac{1}{2}} - \Phi_{\ell-\frac{1}{2}}$	π/LMAX, a constant
Q	None	Temporary storage
QP	None	Temporary storage
SINLP(L)	None	Sin (PP*L)
SMU(M)	$\sqrt{1-\mu_m^2}$	SIN(θ_m)

SUMB(I, LP)	None	Given M, J, I, LP, in SUBROUTINE INWARD: SUMB(I, LP)=SUMB(I, LP) $+\sum_{L=1,L2}$ ETABI(L, LP) $+\sum_{L=1,L2}$ ETABT(L, LP)
SUMO(I, LP)	None	Given M, J, I, LP, in SUBROUTINE INWARD: SUMO(I, LP)=SUMO(I, LP) $+\sum_{L=1,L2}$ ETAOI(L, LP) $+\sum_{L=1,L2}$ ETAOT(L, LP)
TANN	τ	TAN θ_m
TOLER	None	Relative error criterion imposed upon each integration performed by SUBROUTINE SINT (in decimal fraction) (input on card in DRAW)
W(M)	$\frac{\lvert\Delta\mu_m\rvert}{2} = \frac{\lvert\mu_{m+\frac{1}{2}} - \mu_{m-\frac{1}{2}}\rvert}{2}$	$\lvert\cos(M*\Delta\theta) - \cos(M-1)*\Delta\theta)\rvert * \frac{1}{2}$
X(I)	r_i	Outer radius of radial hydro zones (I, J), J=1, JMAX (in cm)
XH(L)	None	Given M, J, I, L: The radial coordinate of the intersection of the characteristic ray of fan L and the "nearest zone face" (see DH(L)) (in cm) (part of "DRAW data" transferred to TRAN2)
XMAX	r_i	=X(I), the outer radius of the radial hydro zone currently being considered (transferred to SUBROUTINES BOT and OUT) (in cm)

XMIN	r_{i-1}	=X(I-1), the inner radius of the radial hydro zone currently being considered (transferred to SUBROUTINES BOT and OUT) (in cm)
XV(L)	None	Given M, J, I, L: The radial coordinate of the intersection of the characteristic ray of fan L and the "nearest zone surface" (see DV(L)) (in cm) (part of "DRAW data" transferred to TRAN2)
YH(L)	None	Given M, J, I, L: $Y(J-\frac{1}{2})$ - (the axial coordinate corresponding to XH(L)) (in cm) (part of "DRAW data" transferred to TRAN2)
YV(L)	None	Given M, J, I, L: $Y(J-\frac{1}{2})$ - (the axial coordinate corresponding to XV(L)) (in cm) (part of "DRAW data" transferred to TRAN2)

FLOW OF CONTROL IN SUBROUTINE OUTWRD

SUBROUTINE OUTWRD is called by SUBROUTINE DRAW to compute, given polar angle θ_M, the η's that couple for face S_1 of zone (I, J) all the fans LP on S_1 containing rays entering the zone traveling in an inward or outward direction with the fans L (on all the other zone faces) which contain rays leaving zone (I, J) traveling in an outward direction. (See SUBROUTINE INWARD for the definition of a ray traveling inward or outward at a point P.)

Given below are the η's that meet the above restriction for the θ_M's (all $< \pi/2$) considered in the "DRAW subroutine sequence," and which are calculated in this routine:

$\eta_{I \to O}$ (L, LP)	L=L2P1, LMAX; LP=L2P1, LMAX Considering rays entering the zone inside face (must be traveling outward) and exiting the zone outside face (must be traveling outward).
$\eta_{I \to T}$ (L, LP)	L=L2P1, LMAX; LP=L2P1, LMAX Considering rays entering the zone inside face and exiting the zone top face traveling outward.
$\eta_{B \to O}$ (L, LP)	L=L2P1, LMAX; LP=1, LMAX Considering rays entering the zone bottom face traveling inward and outward, and exiting the zone outside face.
$\eta_{O \to O}$ (L, LP)	L=L2P1, LMAX; LP=1, L2 Considering rays entering the zone outside face and exiting the zone outside face.
$\eta_{B \to T}$ (L, LP)	L=L2P1, LMAX; LP=1, LMAX Considering rays entering the zone bottom face traveling inward and outward, and exiting the zone top face traveling outward.
$\eta_{O \to T}$ (L, LP)	L=L2P1, LMAX; LP=1, L2 Considering rays entering the zone outside face and exiting the zone top face traveling outward.

The computed η's along with the other "DRAW data" (see glossary) are stored in storage buffer array BUFF1 as they are calculated. At the end of the outward sweep of zone (I,J), BUFF1 is emptied into the output storage buffer array BUFF2 to be saved for output to the "DRAW data" file. If BUFF2 cannot accommodate the BUFF1 load of new data, it is output to the "DRAW data" file. BUFF1 must be able to store all the "DRAW data" for the outward sweep through zone (I,J): if it cannot, the problem is stopped.

The η's for the rays entering the inside zone face can be checked in SUBROUTINE OUTWRD to see if they sum to 1.0 within CHKSUM (see SUBROUTINE DRAW), since rays entering a zone through the inside face can never exit the zone traveling in an inward direction. Preliminary calculations are performed for the sum-check on the remaining η's (see SUBROUTINE DRAW).

Glossary of Variables Used in SUBROUTINE OUTWRD

FORTRAN Label	Report Label	Description
AA(L)	A_ℓ	See INWARD glossary
AID	None	Temporary storage
ANGL1	None	Temporary storage for information transferred to SUBROUTINES IN, OUT, and BOT
ANGL2	None	Same as above
ASN	None	Given I, ASN - arcsin $\frac{X(I-1)}{X(I)}$
BUFF1	None	≡CAP Storage buffer array (see INWARD Glossary)
BUFF2	None	≡P. BUFF2(1) ≡ NOO Output storage array (see DRAW Glossary)
CH(LP)	k_q	= JB(J-1,I,LP,M) (in ergs/sec) (part of "DRAW data" transferred to TRAN2)
CHKSUM	None	Error criterion for the sum of the ETA's (see DRAW Glossary) (input on card in DRAW)

CVIN(LP)	k_q	= JO(J,I-1,LP,M) (in ergs/sec) (part of "DRAW data" transferred to TRAN2)
CVOUT(L)	k_q	= JO(J,I,L,M) (in ergs/sec) (part of "DRAW data" trasnferred in TRAN2)
DBGPRT	None	The debug print control: DBGPRT $\geq 10^{-20}$ means debug print desired; otherwise, debug print not desired
DELTY	$h=z_j - z_{j-1}$	DY(J) (in cm)
DH(L)	t	Given M,J,I,L: The length of the characteristic ray of fan L measured the top face Y(J) of zone (I,J) (see SUBROUTINE DISTNC) (in cm) (part of "DRAW data" transferred in TRAN2)
DQ	None	Temporary storage
DV(L)	t	Civen M,J,I,L: The length of the characteristic ray of fan L measured on vertical face X(I) of zone (I,J) (see SUBROUTINE DISTNC) (in cm) (part of "DRAW data" transferred to TRAN2)
DX(I)	$r_i - r_{i-1}$	Radial zone length of zones (I,J),J=1, JMAX (in cm)
DY(J)	$h=z_j - z_{j-1}$	Vertical length of hydro zones (I,J). I=1,IMAX (in cm)

EPS	None	Absolute error criterion imposed upon each integration performed by SUB-ROUTINE SINT
ETABO(L, LP)	η	Given M, J, I, L, LP: The fraction of (energy in fan LP passing into zone (I, J) through its bottom face in 1 sec) that passes out zone (I, J) through its outside face in fan L, in direction MU(M) in the case of isotropic, homogeneous radiation σ=S=0 (part of "DRAW data" transferred to TRAN2)
ETABT(L, LP)	η	See INWARD glossary (part of "DRAW data" transferred to TRAN2)
ETAIO(L, LP)	η	Given M, J, I, L, LP: The fraction of (energy in fan LP passing into zone (I, J) through its inside face in 1 sec) that passes out zone (I, J) through its outside face in fan L, in direction MU(M) in the case of isotropic, homogeneous radiation σ=S=0 (part of "DRAW data" transferred to TRAN2)
ETAIT(L, LP)	η	Given M, J, I, L, LP: The fraction of (energy in fan LP passing into zone (I, J) through its inside face in 1 sec) that passes out zone (I, J) through its top face in fan L, in direction MU(M) in the case of isotropic, homo-geneous radiation σ=S=o (part of "DRAW data" transferred to TRAN2)

ETAOO(L, LP)	η	Given M, J, I, L, LP: The fraction of (energy in fan LP passing into zone (I, J) through its outside face in 1 sec) that passes back out zone (I, J) through its outside face in fan L, in direction MU(M) in the case of isotropic, homogeneous radiation σ=S=0. (part of "DRAW data" transferred to TRAN2)
ETAOT(L, LP)	η	See INWARD glossary (part of "DRAW data" transferred to TRAN2)
FORPI	4π	4π
HELP1	None	Temporary storage for information transferred to SUBROUTINES IN and BOT
HELP2	None	Temporary storage for information transferred to SUBROUTINE BOT
I	i	Index of the current radial hydro zone with outer radius X(I)
IAID	None	Flag generated for SUBROUTINES BOT and OUT
ICHECK	None	An indicator: ICHECK = 0 means for the current M and J, the "inside" η's sum to 1. ICHECK = 1 means for the current M and J, the "inside η's do not sum to 1. (see SUM)
ISEND	None	Flag generated for SUBROUTINES BOT and OUT
ISENDD	None	Flag for SUBROUTINE DISTNC
ISWEEP	None	Current direction of DRAW sweep: = 1 means sweeping inward = 2 means sweeping outward
J	j	Axial index of the current hydro zone with upper boundary Y(J)

KK1	None	Index of the last current entry in the storage buffer array BUFF1 (must not exceed 1023)
KK2	None	Index of the last current entry in the output storage array BUFF2
KKK	None	A running index
L	ℓ	Fan index
L2	None	LMAX/2
L2P1	None	L2+1
LD	None	A running index.
LHI	None	See INWARD glossary.
LIM1	None	The lower limit in some integrations by SUBROUTINE SINT (in cm)
LIM2	None	The upper limit in some integrations by SUBROUTINE SINT (in cm)
LLOW	None	See INWARD glossary.
LMAX	ℓ_{max}	The total number of fans of directions to be considered in the "DRAW subroutine sequence". (input on card in DRAW)
LMDA	$h\tau$	Given M and J: The horizontal distance a ray will travel in horizontal zone layer DY(J) between intersections with surface Y(J-1) and Y(J) in direction θ_m; or, DY(J) * TAN(θ_m) (in cm)
LP	ℓ'	Fan index.
M	m	Index of the current polar angle θ_m (arccos MU(M)) being considered in constructing the η's.
MU(M)	μ_m	cos θ_m. (See DRAW glossary).

NOBUFF	None	The number of output buffer (BUFF2) loads filled per inward and outward sweep of DRAW along one horizontal layer of zones (J) in direction μ_m (Not a function of M or J)
NOO	None	BUFF2(1) The number of inward and/or outward zone sweeps contained in the current output buffer load BUFF2. (Each buffer load must contain at a minimum one complete (inward or outward) zone sweep). (part of "DRAW data" transferred to TRAN2)
PI	π	π.
PP	$\Delta\Phi = P$ $= \phi_{\ell+\frac{1}{2}} - \phi_{\ell-\frac{1}{2}}$	π/ LMAX, a constant.
Q	None	Temporary storage
QP	None	Temporary storage
R	None	Temporary storage
RP	None	Temporary storage
SINLP(L)		Sin (PP*L)
SMU(M)	$\sqrt{1-\mu_m^2}$	Sin (θ_m)
SUM	None	In checking the sums of the "inside η's," given M, J, I, LP:

$$\text{SUMOO} = \sum_{L=L2P1,\,LMAX} \text{ETAIO(L, LP)}$$

$$\text{SUMTT} = \sum_{L=L2P1,\,LMAX} \text{ETAIT(L, LP)}$$

$$\text{SUM} = \text{SUMOO} + \text{SUMTT}$$

(should = 1.)

SUMB(I, LP)	None	Given M, J, I, LP, in SUBROUTINE OUTWRD: $$SUMB(I,LP)=SUMB(I,LP) + \sum_{L=L2P1,LMAX} ETABO(L,LP) + \sum_{L=L2P1,LMAX} ETABT(L,LP)$$
SUMO(I, LP)	None	Given M, J, I, LP, in SUBROUTINE OUTW $$SUMO(I,LP)=SUMO(I,LP) + \sum_{L=L2P1,LMAX} ETAOO(L,LP) + \sum_{L=L2P1,LMAX} ETAOT(L,LP)$$
SUMOO	None	See SUM
SUMTT	None	See SUM
TANN	τ	Tan (θ_m)
TOLER	None	Relative error criterion imposed upon each integration performed by SUBROUTINE SINT (in decimal fraction) (input on card in DRAW)
W(M)	$\frac{\lvert\Delta\mu_m\rvert}{2} = \frac{\lvert\mu_{m+\frac{1}{2}} - \mu_{m-\frac{1}{2}}\rvert}{2}$	$\lvert\cos(M*\Delta\theta) - \cos((M-1)*\Delta\theta)\rvert * \frac{1}{2}$
X(I)	r_i	Outer radius of radial hydro cells (I, J), J=1, JMAX (in cm)

XH(L)	None	Given M, J, I, L: The radial coordinate of the intersection of the characteristic ray of fan L and the "nearest zone face" (see DH(L)) (in cm) (part of "DRAW data" transferred to TRAN2)
XI	None	Temporary storage
XMAX	r_i	= X(I), the outer radius of the radial hydro zone currently being considered (transferred to SUBROUTINES IN, OUT, and BOT) (in cm)
XMIN	r_{i-1}	= X(I-1), the inner radius of the radial hydro zone currently being considered (transferred to SUBROUTINES IN, OUT and BOT) (in cm)
XV(L)	None	Given M, J, I, L: The radial coordinate of the intersection of the characteristic ray of fan L and the "nearest zone surface" (see DV(L)) (in cm) (part of "DRAW data" transferred to TRAN2)
YH(L)	None	Given M, J, I, L: $Y(J-\frac{1}{2})$ - the axial coordinate corresponding to XH(L) (in cm) (part of "DRAW data" transferred to TRAN2)
YV(L)	None	Given M, J, I, L: $Y(J-\frac{1}{2})$ - the axial coordinate corresponding to XV(L) (in cm) (part of "DRAW data" transferred to TRAN2)

FLOW OF CONTROL IN SUBROUTINE DISTNC

Given a range of fans(LIM1, LIM2), the current polar angle θ_m, the direction of the current sweep in SUBROUTINE DRAW, and the current zone (I, J) under consideration in SUBROUTINE DRAW, SUBROUTINE DISTNC calculates the "length, D, of the characteristic ray" of the specified fans in zone (I, J) and the coordinates of each ray's intersection with "the nearest zone face,"

Given zone (I, J), the following definitions are in order:

A. Fans measured on horizontal face Y(J)

The "characteristic ray" of fan L is a ray measured from the midpoint of the top face of zone (I, J), [X(I-1/2), Y(J)], in a direction specified by azimuthal angle $\Phi = \Phi_{LMAX+1-L}$ and polar angle $\theta = \pi$ - arccos [MU(M)] (see Fig. 13.)

The "length, D, of the characteristic ray" is the distance from [X(I-1/2), Y(J)] in the above specified direction to the "nearest zone face", which could be the outside face of zone (I, J), the bottom face of zone (I, J) or the inside face of zone (I, J).

B. Fans measured on vertical face X(I-1) or X(I)

1) Sweeping outward, L > L2 [Fans measured on face X(I)].

The "characteristic ray" of fan L is a ray measured from the midpoint of the outside face of zone (I, J), [X(I), Y(J-1/2)], in a direction specified by azimuthal angle $\Phi = \Phi_{LMAX+1-L}$ and polar angle $\theta = \pi$ - arccos [MU(M)] (See Fig. 14.)

The "length, D, of the characteristic ray" is the distance from [X(I), Y(J-1/2)] in the above specified direction to the "nearest zone face," which could be the bottom face of zone (I, J), the inside face of zone (I, J), or the outside face of zone (I, J).

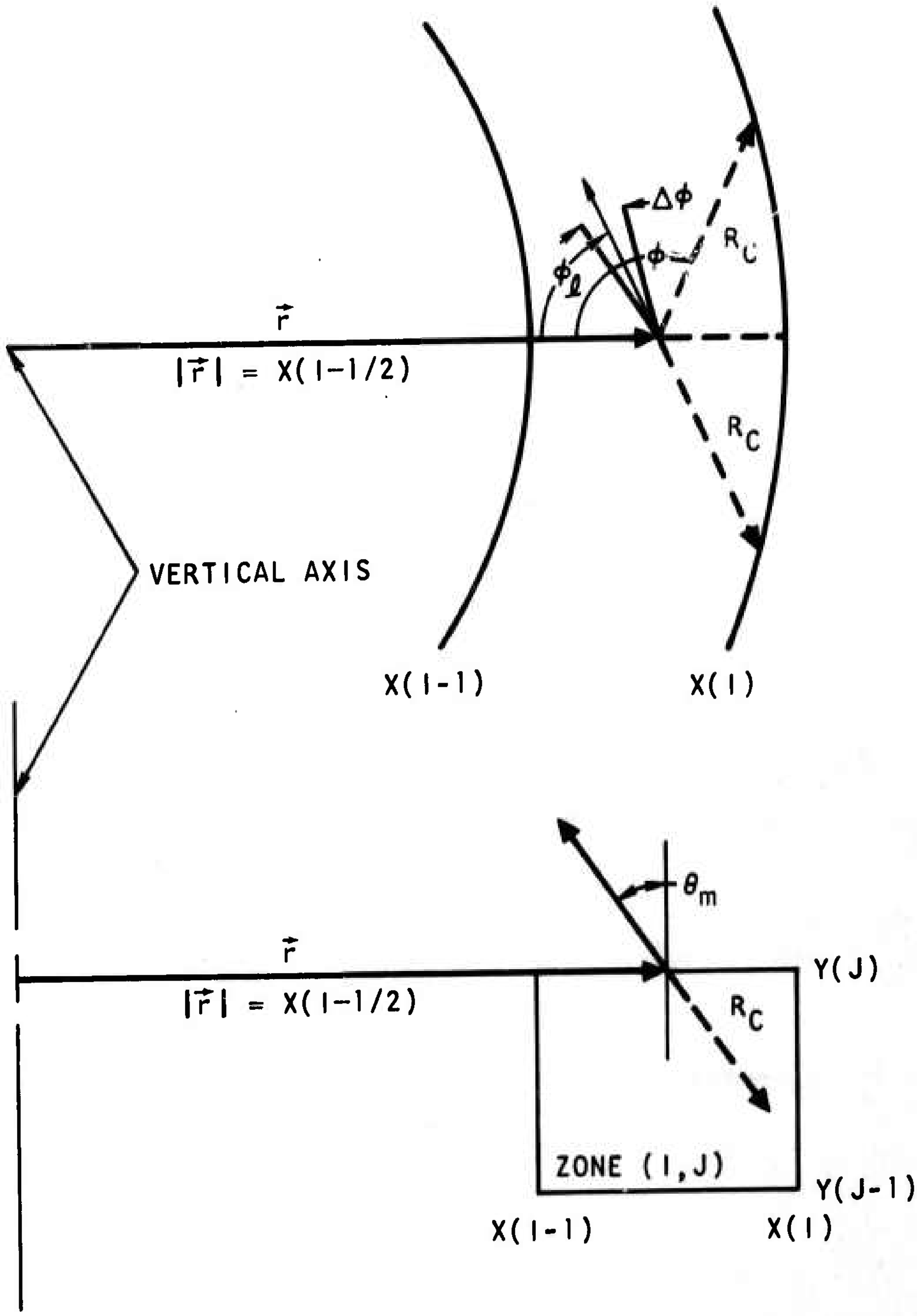


Figure 13. Characteristic Ray, R_c, for Fans Measured on Horizontal Face Y(J) of Zone (I,J)

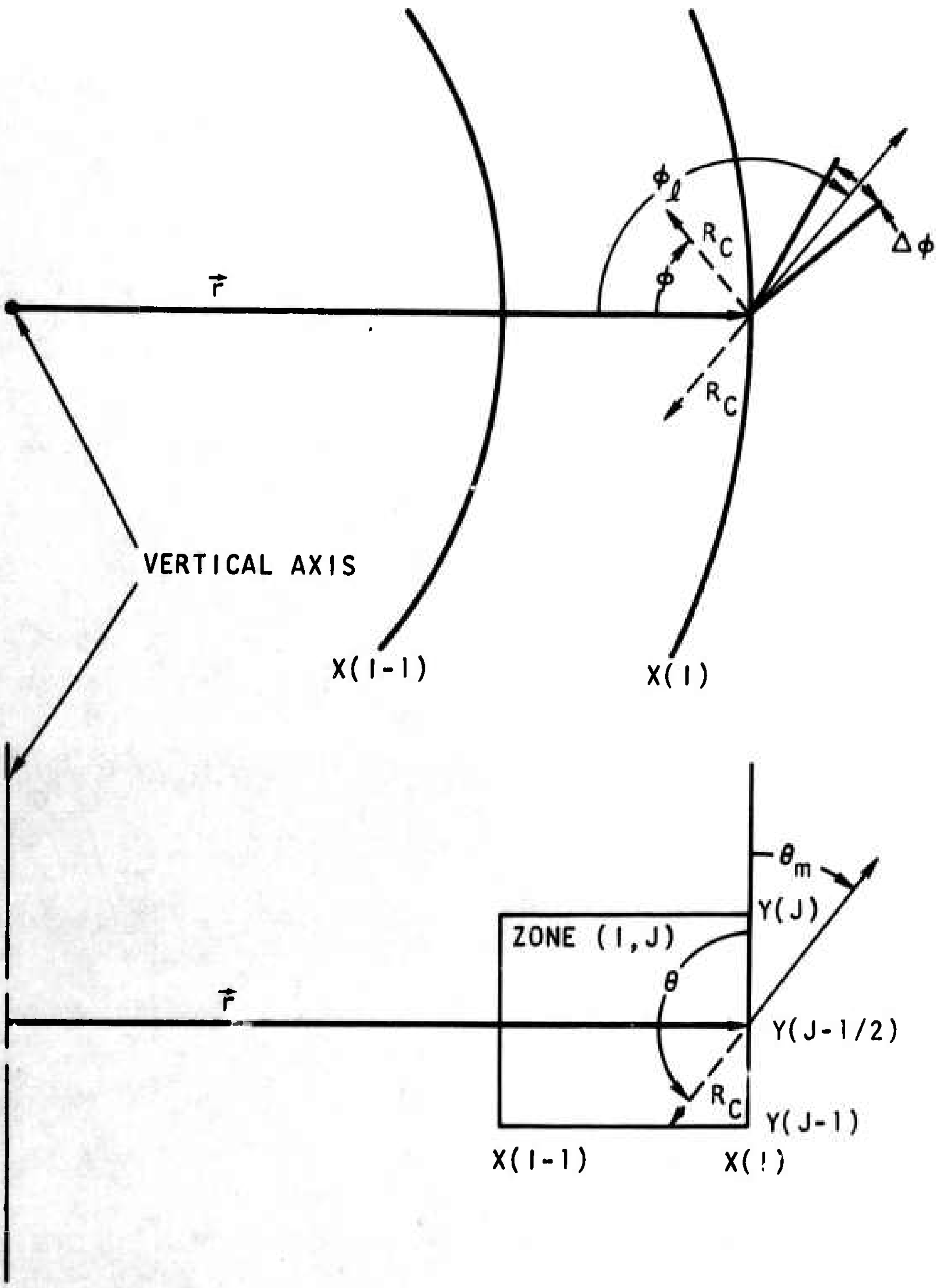


Figure 14. Characteristic Ray, R_c, for Fans Measured on Vertical Face X(I) of Zone (I, J)

2) Sweeping inward, $L \leq L2$ [fans measured on face X(I-1)].
The "characteristic ray" of fan L is a ray measured from the midpoint of the inside face of zone (I,J), [X(I-1), Y(J-1/2)], in a in a direction specified by azimuthal angle $\Phi = \Phi_{LMAX+1-L}$ and polar angle $\theta = \pi$ - arccos [MU(M)] (see Fig. 15).

The "length, D, of the characteristic ray" is the distance from [X(I-1), Y(J-1/2)] in the above specified direction to the "nearest zone face," which could be the bottom face of zone (I,J) or the outside face of zone (I,J).

In the coding, the projection of D in the horizontal plane containing the characteristic ray's origin is first calculated, and then the three-dimensional length is computed.

NOTE: The coordinates (x,y) of the intersection of the "characteristic ray" and "the nearest zone face" are calculated; however, y' = [Y(J-1/2)-y] is returned to SUBROUTINES INWARD and OUTWRD instead of th axial coordinate, y. This feature is to enable SUBROUTINE TRAN2 to reconstruct the axial coordinate y_{TRAN2} for polar directions $\theta \leq \pi/2$ (in which case y_{TRAN2} = Y(J-1/2)-y') and $\theta > \pi/2$ (in which case y_{TRAN2} = Y(J-1/2) + y') using the restrictions that the θ_M, M=1, MUMAX, must be symmetric about $\theta = \pi/2$.

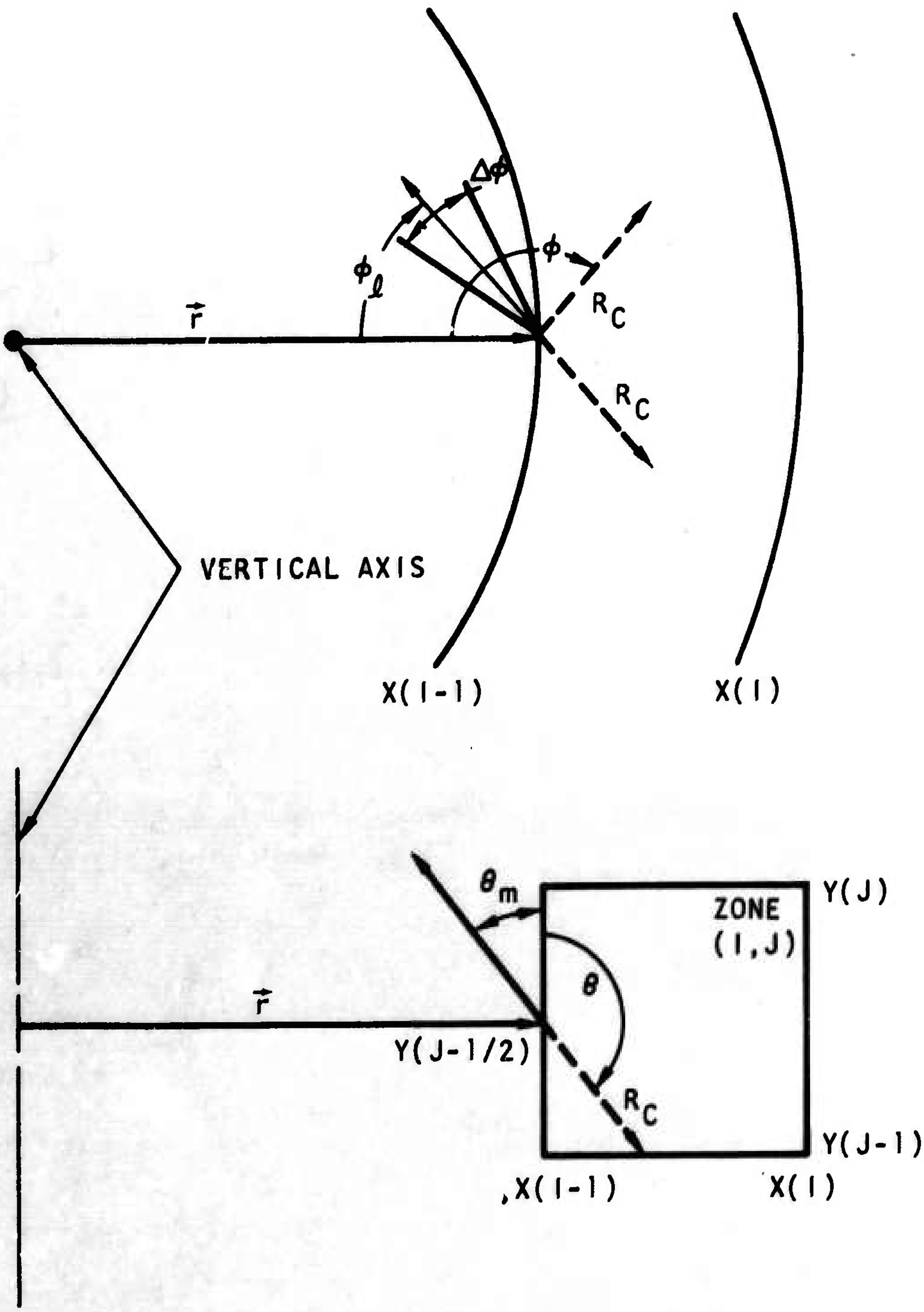


Figure 15. Characteristic Ray, R_c, for Fans Measured on Vertical Face X(I-1) of Zone (I, J)

Glossary of Variables Used in SUBROUTINE DISTNC

FORTRAN Label	Report Label	Description
A	None	Temporary storage.
B	None	Temporary storage.
BBB	None	Temporary storage.
COSPHI(L)	$\cos\Phi_\ell = \cos\left[(\ell - \frac{1}{2})\Delta\Phi\right]$	Cos (PHI(L)).
DBGPRT	None	The debug print control: DBGPRT $\geq 10^{-20}$ means debug print desired; otherwise, debug print not desired.
DH(L)	t	Given M, J, I, L: The length of the characteristic ray of fan L measured on the horizontal face Y(J) of zone (I, J). (in cm)
DHH	None	Given M, J, I, L: The projection of DH(L) onto plane Y(J). (in cm)
DV(L)	t	Given M, J, I, L: The length of the characteristic ray of fan L measured on vertical face X(I) or X(I-1) of zone (I, J). (in cm)
DVV	None	Given M, J, I, L: The projection of DV(L) onto plane $Y(J-\frac{1}{2})$ (in cm)
DX(I)	$r_i - r_{i-1}$	Radial zone length of hydro zones (I, J), J=1, JMAX (in cm)

DY(J)	$h = z_j - z_{j-1}$	Vertical length of hydro zones (I, J), J=1,JMAX (in cm)
HELP	None	Temporary storage
HIT	None	In computing characteristic ray lengths, the first vertical zone boundary pierced by the current characteristic ray (in cm)
I	None	In calculating the lengths of characteristic rays originating from vertical zone faces: I=II-1 if current sweep is inward I=II if current sweep is outward such that X(I) is the face from which the current characteristic ray is measured. If calculating the characteristic ray lengths originating from horizontal zone faces: I=II
IBACK	None	Flag used in computing XV(L)
II	i	Index of the current radial hydro zone with outer radius X(II) (I of SUBROUTINE INWARD or OUTWRD)
IMAX	i_{max}	The number of radial hydro zones
ISENDD	None	A flag: = 1 means compute lengths of characteristic rays originating from a vertical zone face = 2 means compute lengths of characteristic rays originating from a horizontal zone face
ISWEEP	None	Current direction of DRAW sweep: = 1 means sweeping inward = 2 means sweeping outward
J	j	Index of the current axial (vertical) hydro zone with upper boundary Y(J)
K	None	A running index

L	ℓ	Fan
L2	None	LMAX/2
LIM1	None	= LLOW in SUBROUTINES INWARD and OUTWRD = complement of highest index of the fans for which characteristic ray lengths are to be calculated
LIM1P	None	An index limit used in a debug print (complement of LIM2)
LIM2	None	= LHI in SUBROUTINES INWARD and OUTWRD = complement of the lowest index of the fans for which characteristic ray lengths are to be calculated
LIM2P	None	An index limit used in a debug print (complement of L1M1)
LMAX	ℓ_{max}	The total number of fans of directions to be considered in the "DRAW subroutine sequence."
M	m	Index of the current polar angle θ_m (arccos MU(M)) being considered in constructing the η's.
MU(M)	m	Cos θ_m. (See DRAW glossary.)
PHI(L)	Φ_ℓ	Φ_ℓ. (See DRAW glossary.)
SMU(M)	$\sqrt{1-\mu_m^2}$	Sin (θ_m)
X(I)	r_i	Outer radius of radial hydro zones (I, J), J=1, JMAX (in cm)
XH(L)	None	Given M, J, I, L: The radial coordinate of the intersection of the characteristic ray of fan L and the "nearest zone face" (see DH(L)) (in cm)

XV(L)	None	Given M, J, I, L: The radial coordinate of the intersection of the characteristic ray of fan L and the "nearest zone face" (see DV(L)) (in cm)
YH(L)	None	Given M, J, I, L: $Y(J-\frac{1}{2})$ - [the axial coordinate corresponding to XH(L)] (in cm)
YV(L)	None	Given M, J, I, L: $Y(J-\frac{1}{2})$ - [the axial coordinate corresponding to XV(L)] (in cm)

FLOW OF CONTROL IN FUNCTION JO(J, I, L, M)

Given the vertical cylindrical surface X(I) of height DY(J), the polar angle θ_m (arccos MU(M)) with range $\Delta\mu_m$ (2*W(M)), and fan L with range $\Delta\Phi_\ell$, FUNCTION JO calculates the rate at which energy is passing through surface X(I) of area DY(J)*2*π*X(I) in fan L in direction MU(M) in the case of isotropic, homogeneous radiation σ=S=0 divided by I_o, the intensity of the radiation field (or, equivalently, with I_o = 1) (See Fig. 16):

$$JO = \frac{1}{I_o} \int_{Y(J-1)}^{Y(J)} \int_{(L-1)*PP}^{(L)*PP} I_o * 2 * \pi * X(I) * \sin\theta_m * |\cos\Phi_\ell| \; \Delta\mu_m d\Phi dz$$

$$= X(I)*\sin\theta_m * 4 * \pi * W(M) * DY(J)*2*\sin(\frac{PP}{2}) \left|\cos\left[L-\frac{1}{2}\right]\right|$$

(See Eq. 151.)

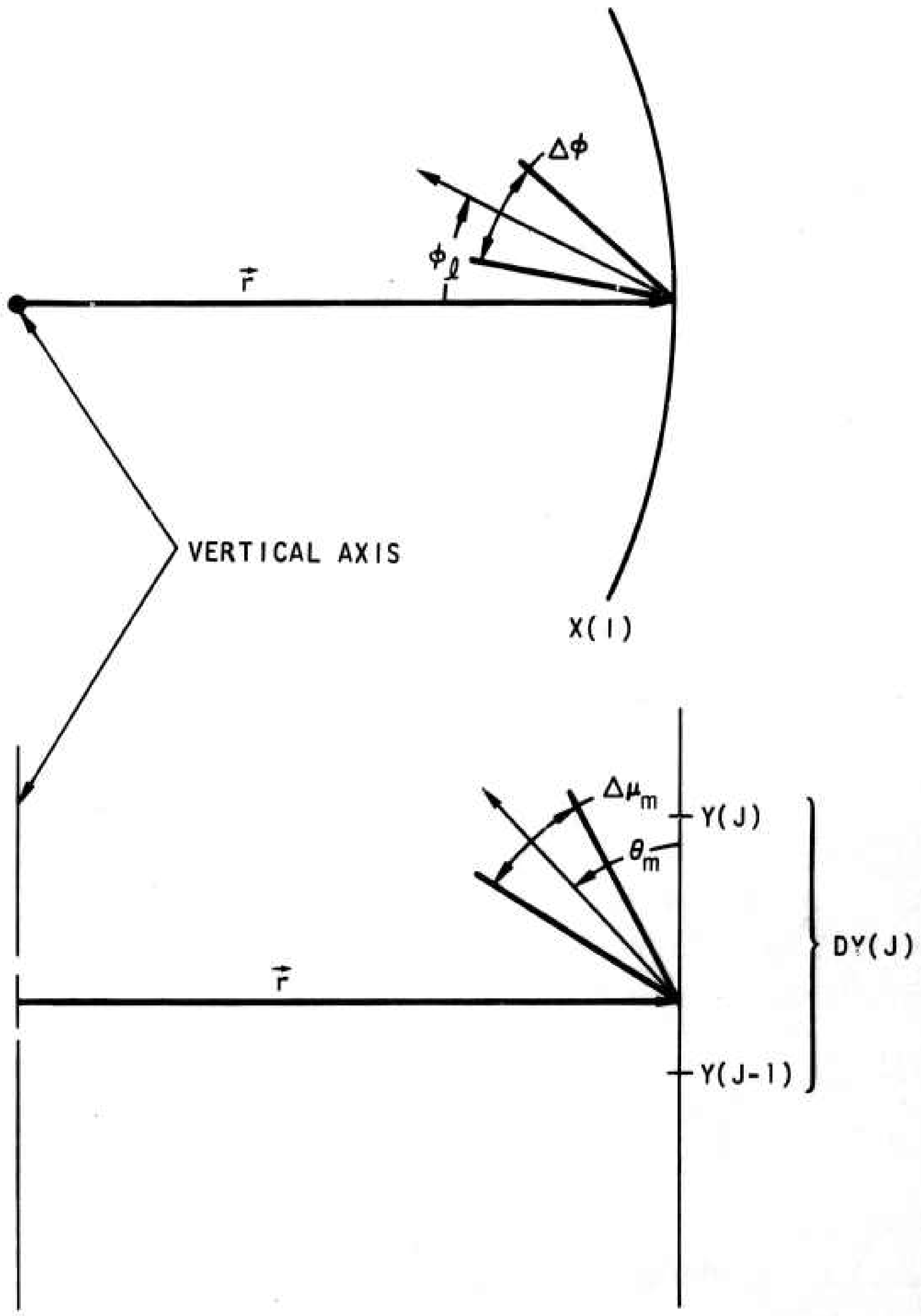


Figure 16. Fan of Directions for FUNCTION JO

Glossary of Variables Used in FUNCTION JO(J,I,L,M)

FORTRAN Label	Report Label	Description
COSPHI(L)	$\cos\Phi_\ell$	cos (PHI(L))
	$= \cos\left[(\ell-\frac{1}{2})\Delta\Phi_\ell\right]$	
DY(J)	h	Y(J) - Y(J-1) (in cm)
FORPI	4π	4π
I	i	Index of the current radial hydro zone with outer radius X(I)
J	j	Index of the current axial (vertical) hydro zone with upper boundary Y(J)
M	m	Index of the current polar angle θ_m (arccos MU(M) being considered
SINP2	$\sin\frac{\Delta\Phi_\ell}{2}$	$\sin\left(\frac{PP}{2}\right)$
SMU(M)	$\sqrt{1-\mu_m^2}$	$\sin(\theta_m)$
W(M)	$\frac{\lvert\Delta\mu_m\rvert}{2}$	$\lvert\cos[M*\Delta\theta] - \cos[(M-1)*\Delta\theta]\rvert * \frac{1}{2}$
	$= \frac{\lvert\mu_{m+\frac{1}{2}} - \mu_{m-\frac{1}{2}}\rvert}{2}$	
X(I)	r_i	Outer radius of radial hydro zones (I,J), J=1,JMAX (in cm)

FLOW OF CONTROL IN FUNCTION JB(J,I,L,M)

Given horizontal annulus DX(I) of area $\pi[X^2(I)-X^2(I-1)]$, the polar angle θ_m (arccos MU(M))with range $\Delta\mu_m$ (2*W(M)), and fan L with range $\Delta\Phi_\ell$, FUNCTION JB calculates the rate at which energy is passing through annulus DX(I) in fan L in direction MU(M) in the case of isotropic,

homogeneous radiation σ=S=0 divided by I_o, the intensity of the radiation field (or, equivalently, with I_o = 1) (see Fig. 17):

$$JB = \frac{1}{I_o} \int_{X(I-1)}^{X(I)} \int_{(L-1)*PP}^{(L)*PP} I_o * 2 * \pi * r * |\mu_m| \, \Delta\mu_m \, d\Phi dr$$

if A_ℓ = X(I) * SIN(L*PP)

and $A_{\ell-1}$ = X(I) * SIN ((L-1)*PP)

$$= 4*\pi*|\mu_m|*W(M)$$

$$* \left\{ \left[\frac{r^2}{2} \arcsin\left(\frac{A_\ell}{r}\right) + \frac{A_\ell}{2} \sqrt{r^2 - A_\ell^2}\right]_{X(I-1)\vee A_\ell}^{X(I)} \right.$$

$$- \left[\frac{r^2}{2} \arcsin\left(\frac{A_{\ell-1}}{r}\right) + \frac{A_{\ell-1}}{2} \sqrt{r^2 - A_{\ell-1}^2}\right]_{X(I-1)\vee A_{\ell-1}}^{X(I)}$$

$$\left. + \frac{\pi}{2} \left[\frac{r^2}{2}\right]_{X(I-1)\vee A_{\ell-1}}^{X(I-1)\vee A_\ell} \right\}$$

(See Eq. 152.)

NOTE: Since the Φ_ℓ are symmetric about $\Phi = \pi/2$, JB(J,I,L,M) = JB(J,I,LMAX+1-L,M).

Glossary of Variables Used in FUNCTION JB (J,I,L,M)

FORTRAN Label	Report Label	Description
A	None	Dummy variable for ARITHMETIC STATEMENT FUNCTION I1(A,R)

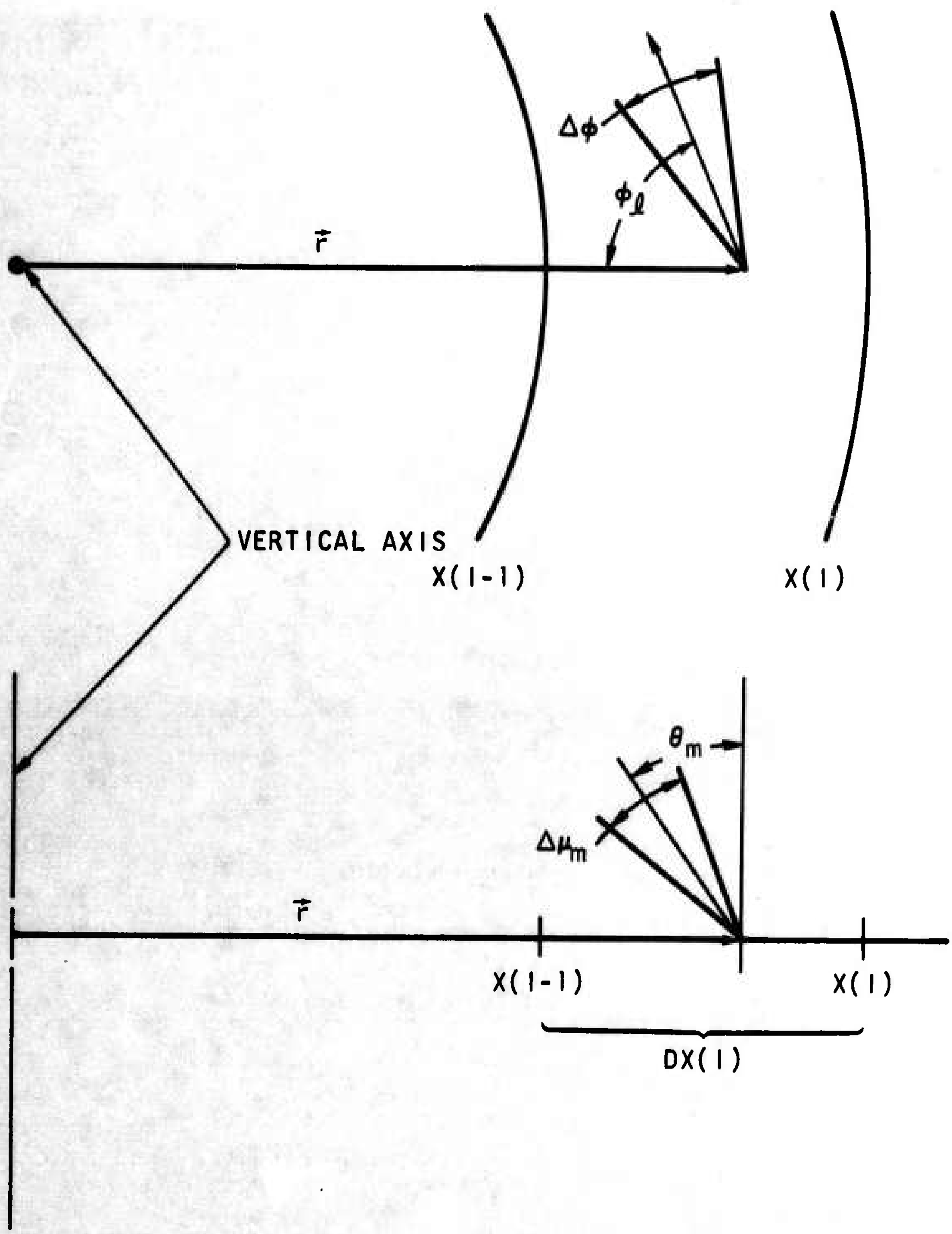


Figure 17. Fan of Directions for FUNCTION JB

AA(L)	a_ℓ	At a given radius X(I): AA(L) = SINLP(L)*X(I), the distance of closest approach (to the vertical axis) of a horizontal ray making an azimuthal angle (π-L*PP) with a radius vector at radius X(I) (in cm)
FORPI	4π	4π
I	i	Index of the current radial hydro zone with outer radius X(I)
I1	None	ARITHMETIC STATEMENT FUNCTION
L	ℓ	Fan index
L2	None	LMAX/2
LIM1	None	Temporary storage
LIM2	None	Temporary storage
LL	None	Index used in calculation of JB
LMAX	ℓ_{max}	The total number of fans being considered
M	m	Index of the current polar angle θ_m (arccos MU(M)) being considered
MU(M)	$\mu_m = \cos\theta_m$	$\cos(\theta_m)$
PI4	$\pi/4$	$\pi/4$
W(M)	$\left\lvert\frac{\Delta\mu_m}{2}\right\rvert = \frac{\lvert\mu_{m+\frac{1}{2}}\ \mu_{m-\frac{1}{2}}\rvert}{2}$	$\lvert\cos[M*\Delta\theta] - \cos[(M-1)*\Delta\theta]\rvert * \frac{1}{2}$
X(I)	r_i	Outer radius of radial hydro zones (I,J), J=1,JMAX (in cm)

FUNCTION SINT (A1,B1,EPS1,FUNC)

FUNCTION SINT numerically evaluates the definite integral I,

$$I = \int_{x=A1}^{x=B1} \text{"FUNC"}\, dx$$

according to Simpson's method prescribed by the COMMUNICATIONS OF THE ACM ALGORITHM 182 (Ref. 7) where:

A1 = the lower limit of integration

B1 = the upper limit of integration

EPS1 = the accuracy criterion in the units of the variable of integration, x.

"FUNC" = the name of the FORTRAN FUNCTION routine which evaluates the desired integrand as a function of x.

FLOW OF CONTROL IN FUNCTION IN(Z)

FUNCTION SINT is called from SUBROUTINE OUTWRD to evaluate an η integral I of the form

$$I = \int_{Z_1}^{Z_2} [\text{INTEGRAND}]\, dZ$$

and FUNCTION IN is in turn called by SINT to evaluate the INTEGRAND (which is a function of Z) several times during its evaluation of I. The INTEGRAND is of the form:

$$\text{INTEGRAND} = \sin\,[A \vee Y]$$

where $A = PP*(LP-1)$

or

$$A = \pi - \arcsin\left[X(I) - \frac{\sin\,(L-1)PP}{X(I-1)}\right]$$

and

γ = a limiting angle (see GAM in glossary)

Glossary of Variables Used in FUNCTION IN(Z)

FORTRAN Label	Report Label	Description
ANGL1	None	= PP * (LP-1), the angle $\Phi_{\ell-1}$ measured at the inside of the current zone, or surface X(I-1) or $= \pi$ - ASIN $\left[\frac{X(I)*SIN LP(L-1)}{X(I-1)}\right]$, the angle $\Phi_{\ell-1}$ measured at the inside of the current zone, or surface X(I-1) (in radians) (value received from SUBROUTINE OUTWRD)
DELTY	h	DY(J)
GAM	γ_o	The largest azimuthal angle (≥ 0) at Z units above Y(J-1) which a ray entering the inside face X(I-1) of the current zone can make with the radius vector and intersect the top zone face before intersecting the outer zone face (in radians)
HELP1	None	Temporary storage information received from SUBROUTINE OUTWRD
LMDA	$h\tau$	Given J and direction θ_m: The horizontal distance a ray will travel between horizontal plane Y(J-1) + Z and Y(J); or, $\left[DY(J) - Z * TAN(\theta_m)\right]$
PI	π	π
PI2	$\pi/2$	$\pi/2$
TANN	τ	TAN(θ_m)

XMAX	r_i	Outer radius of the radial hydro zone currently being considered (=X(I)) (in cm) (value received from SUBROUTINE OUTWRD)
XMIN	r_{i-1}	Inner radius of the radial hydro zone currently being considered (=X(I-1)) (in cm) (value received from SUBROUTINE OUTWRD)
Z	$h-\zeta$	The distance above Y(J-1) at which the integrands for ETAIO(L, LP) and ETAIT(L, LP) are to be evaluated for FUNCTION SINT (in cm) (value received from FUNCTION SINT)

FLOW OF CONTROL IN FUNCTION BOT(Z)

FUNCTION SINT is called from SUBROUTINES INWARD and OUTWRD to evaluate η integrals and FUNCTION BOT is in turn called by SINT to evaluate the integrands (which are a function of radial distance) for these integrals several times during the course of each integration. The following table specifies each integrand type and relationship to the η calculation in SUBROUTINES INWARD or OUTWRD:

Subroutine in which integral is used	The η associated with the integral	For current fan, direction in which rays ENTER	EXIT	Defining flags in this routine	Equation specifying integrand and integral
INWARD	$\eta_{B\to I}$	IN	IN	ISEND=1	Eq. (158)
INWARD	$\eta_{B\to T}$	IN	IN	ISEND=2 IAID=1	Eq. (171)
OUTWRD	$\eta_{B\to 0}$	IN	OUT	ISEND=2 IAID=4	Eq. (174)
OUTWARD	$\eta_{B\to 0}$	OUT	OUT	ISEND=2 IAID=5	Eq. (175)

Subroutine in which integral is used	The associated with the integral	For current fan, direction in which rays ENTER	EXIT	Defining flags in this routine	Equation specifying integrand and integral
OUTWRD	$\eta_{B \to T}$	IN	OUT	ISEND=2 IAID=2	Eq. (172)
OUTWRD	$\eta_{B \to T}$	OUT	OUT	ISEND=2 IAID=3	Eq. (173)

Glossary of Variables Used in FUNCTION BOT(Z)

FORTRAN Label	Report Label	Description
A	w	In computing the integrands for ETABT (L,LP) and ETABO(L,LP), A= the upper azimuthal angle limit; that is, $$\text{ETA} = \frac{1}{C} \int_{r_o}^{r_f} r\left[(A-B)\vee o.\right] dr$$ and B = the lower azimuthal angle limit.
AAB	None	Temporary storage
ANGL1	None	In computing the integrand for ETABI (L,LP): ANGL1 = X(I) * SINLP(LP), the distance of closest approach of a ray with azimuthal angle ϕ = LP*PP measured at X(I) or = X(I-1) * SINLP(L), the distance of closest approach of a ray with azimuthal angle ϕ = L*PP measured at X(I-1)

		In computing the integrands of ETABI (L, LP) and ETABO(L, LP):
		ANGL1 = X(I) * SINLP(LP)
		(in cm)
ANGL2	None	= X(I) * SINLP(LP-1), the distance of closest approach of a ray with azimuthal angle ϕ = (LP-1)*PP measured at X(I)
		(in cm)
B	α	See A
GAM	γ	In computing the integrand for ETABI (L, LP):
		The largest azimuthal angle ($\geq$0) which a ray entering the bottom face of the current zone at radius Z, X(I-1) $\leq$ Z $\leq$ X(I), can make with the radius vector and intersect the inside zone face before intersecting the outside or top zone faces.
		In computing the integrands for ETABT (L, LP) and ETABO(L, LP):
		Temporary storage
GAM1	γ	See GAM
GAM2	γ_o	The largest azimuthal angle ($\geq$GAM1) that a ray entering the bottom face of the current zone at radius Z, X(I-1) $\leq$ Z $\leq$ X(I), can make with the radius vector and intersect the top zone face before intersecting the outside zone face
HELP	None	Temporary storage
HELP1	None	Temporary storage information received from SUBROUTINES INWARD or OUTWRD
HELP2	None	Same as above
I	i	Index of the current radial hydro zone with outer radius X(I)

IAID	None	Flag: In computing the integrand for ETABT (L, LP): = 1 means rays enter the current zone's bottom face going inward, and exit the zone's top face going inward = 2 means rays enter the current zone's bottom face going inward, and exit the zone's top face going outward = 3 means rays enter the current zone's bottom face going outward, and exit the zone's top face going outward In computing the integrand for ETABO (L, LP): = 4 means rays enter the current zone's bottom face going inward, and exit the zone's outside face going outward = 5 means rays enter the current zone's bottom face going outward, and exit the zone's outside face going outward (value received from SUBROUTINES INWARD or OUTWRD)
ISEND	None	Flag: = 1 means integrand for ETABI (L, LP) desired = 2 means integrands for ETABT (L, LP) or ETABO(L, LP) desired (value received from SUBROUTINE INWARD or OUTWARD)

LMDA	h_τ	Given J and direction θ_m: The horizontal distance a ray will travel in horizontal cell layer DY(J) between intersections with surface Y(J-1) and Y(J); or, $DY(J) * TAN(\theta_m)$ (in cm) (value received from SUBROUTINES INWARD or OUTWRD)
PI	π	π
XMAX	r_i	The outer radius of the radial hydro zone currently being considered (=X(I)) (in cm) (value received from SUBROUTINES INWARD or OUTWRD)
XMIN	r_{i-1}	The inner radius of the radial hydro zone currently being considered (=X(I-1)) (in cm) (value received from SUBROUTINE INWARD or OUTWRD)
Z	r	The radial coordinate, r, at which integrands for ETABI(L,LP), ETABT (L,LP), or ETABO(L,LP) are to be evaluated for FUNCTION SINT (in cm) (value received from FUNCTION SINT)

FLOW OF CONTROL IN FUNCTION OUT(Z)

FUNCTION SINT is called from SUBROUTINES INWARD and OUTWRD to evaluate η integrals and FUNCTION OUT is in turn called by SINT to evaluate the integrands (which are a function of axial distance) for these integrals several times during the course of each integration. The

following table specifies each integrand type and relationship to the η calculation in SUBROUTINES INWARD or OUTWRD:

SUBROUTINE in which integral is used	The η associated with the integral	For current fan, direction in which rays ENTER	EXIT	Defining flags in this routine	Equation specifying integrand and integral
INWARD	$\eta_{0\to I}$	IN	IN	ISEND=1	Eq. 157
INWARD	$\eta_{0\to T}$	IN	IN	ISEND=2 IAID=1	Eq. 159
OUTWRD	$\eta_{0\to T}$	IN	OUT	ISEND=2 IAID=2	Eq. 161
OUTWRD	$\eta_{0\to 0}$	IN	OUT	ISEND=3	Eq. 162

Glossary of Variables Used in FUNCTION OUT(Z)

FORTRAN Label	Report Label	Description
A	w	In computing integrands for ETAOT (L, LP) and ETAOO(L, LP), A = the upper azimuthal angle limit; that is, $$\mathrm{ETA} = \frac{1}{C}\int_{Z_o}^{Z_F} [\sin(A) - \sin(B)]\, v\, 0 \cdot dZ$$ and B = the lower azimuthal angle limit (in radians)
ANGL1	None	In computing the integrand for ETAOI (L, LP): $= \mathrm{PP} * \mathrm{LP},\ \phi_{\ell'}$ or $= \frac{X(I-1)}{X(I)} * \mathrm{SINLP}(L),\ \phi_\ell$ measured at X(I)

		In computing the integrands for ETAOT (L, LP) and ETAOO(L, LP): = PP*(LP-1), $\phi_{\ell'-1}$ (in radians)
ANGL2	None	In computing the integrand for ETAOT (L, LP): = PP * LP, $\phi_{\ell'}$ (in radians)
B	α	See A (in radians)
DELTY	h	DY(J) (in cm)
GAMMO(IG, Z)	γ	IG = 1: = the largest azimuthal angle ($\geq$ 0) which a ray entering Z units above Y(J-1) through the outside face of the current zone can make with the radius vector and intersect the inside zone face before intersecting the top zone face
	γ_o	IG = 2: = the largest azimuthal angle ($\geq$ 0) which a ray entering Z units above Y(J-1) through the outside face of the current zone can make with the radius vector and intersect the top zone face before intersecting the outside zone face (in radians)
IAID	None	Flag: In computing the integrand for ETAOT (L, LP): = 1 means considering rays entering the current zones' outer face going inward, and exiting the zone's top face going inward

		= 2 means considering rays entering the current zone's outer face going inward, and exiting the zone's top face going outward (value received from SUBROUTINES INWARD and OUTWRD)
ISEND	None	Flag: = 1 means integrand for ETAOI (L, LP) desired = 2 means integrand for ETAOT (L, LP) desired = 3 means integrand for ETAOO (L, LP) desired (value received from SUBROUTINES INWARD and OUTWRD)
TANN	τ	TAN(θ_m)
XMAX	r_i	The outer radius of the radial hydro zone currently being considered (=X(I)) (in cm) (value received from SUBROUTINES INWARD and OUTWRD)
XMIN	r_{i-1}	The inner radius of the radial hydro zone currently being considered (= X(I-1)) (in cm) (value received from SUBROUTINES INWARD and OUTWRD)
Z	$h - \zeta$	The distance above Y(J-1) at which the integrands for ETAOI(L, LP), ETAOT (L, LP), and ETAOO(L, LP) are to be evaluated for FUNCTION SINT. (in cm)

FLOW OF CONTROL IN FUNCTION GAMMO (IG, Z)

FUNCTION GAMMO is called by FUNCTION OUT to calculate two limiting angles that are a part of the integrands calculated in OUT. GAMMO returns the value of one of these limiting angles for each call to the FUNCTION. (See description of GAMMO (IG, Z) in OUT glossary.)

Glossary of Variables Used in FUNCTION GAMMO (IG, Z)

FORTRAN Label	Report Label	Description
ASN	None	Arcsin $\left(\frac{X(I-1)}{X(I)}\right)$ (in radians) (Value received from SUBROUTINES INWARDS or OUTWRD)
DELTY	h	DY(J)
HLP	None	Temporary storage
I	i	Index of the current radial hydro zone with outer radius X(I)
IG	None	Flag (See OUT Glossary)
LMDA	$\tau\zeta$	Given J and direction θ_m: The horizontal distance a ray will travel between horizontal plane Y(J-1) + Z and Y(J); or, $(DY(J) - Z) * TAN(\theta_m)$ (in cm)
LMDA2	$\tau^2\zeta^2$	$LMDA^2$
TANN	τ	$TAN(\theta_m)$
XMAX	r_i	Outer radius of the radial hydro zone currently being considered (=X(I)) (in cm)
XMIN	r_{i-1}	Inner radius of the radial hydro zone currently being considered (= X(I-1)) (in cm)

Z	$h - \zeta$	The distance above Y(J-1) at which GAMMO is to be calculated for FUNCTION OUT (in cm) (Value received from FUNCTION OUT which in turn receives it from FUNCTION SINT)

SECTION VII

FLOW OF CONTROL IN SUBROUTINE TRAN2 AND THE "TRANSPORT SUBROUTINE SEQUENCE" FOR A MONO-FREQUENCY, EXPLICIT TEMPERATURE PROBLEM

SUBROUTINE TRAN2 specifies the over-all flow of control through the "transportation subroutine sequence" (shown in Fig. 18). Transport calculations along characteristic ray paths supplied by SUBROUTINE DRAW are performed according to the temperatures and optical depths of the hydro zones and the η's, or "view factors," coupling the energy passing through neighboring faces of these hydro zones.

"Transport calculations" for a given hydro zone (I, J) and a given direction of transport defines the following process:

Given the direction of transport to be in an $\left(\begin{array}{l}\text{inward and upward}\\ \text{outward and upward}\\ \text{inward and downward}\\ \text{outward and downward}\end{array}\right)$

direction, the rate at which energy is leaving the zone through each fan L

on the $\left(\begin{array}{l}\text{inside and top}\\ \text{outside and top}\\ \text{inside and bottom}\\ \text{outside and bottom}\end{array}\right)$ zone faces is calculated from the knowledge

of

a. the contributions to this rate from the rate at which energy is entering the zone through all the fans LP on the

$\left(\begin{array}{l}\text{bottom and outside}\\ \text{inside, bottom, and outside}\\ \text{top and outside}\\ \text{inside, top, and outside}\end{array}\right)$ zone faces assuming an isotropic,

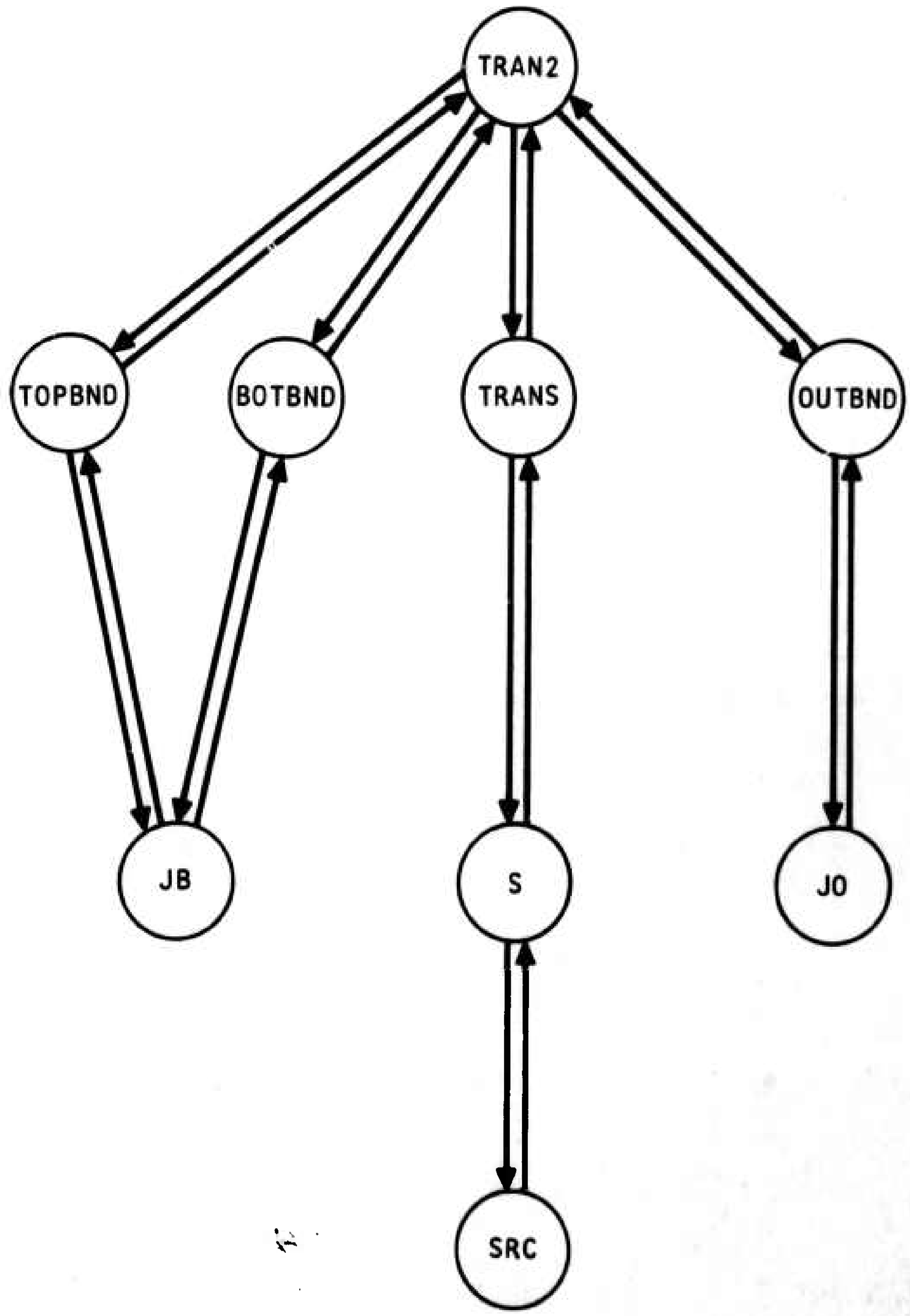


Figure 18. SUBROUTINES of the "Transport Subroutine Sequence"

homogeneous radiation field within the zone. (These contributions are represented by the term $\Sigma\eta \cdot E$ in the transport equation--see SUBROUTINE TRANS.)

and

b. how these contributions are to be modified given a nonzero absorption coefficient and source associated with the zone. (The modification is specified in SUBROUTINE TRANS.)

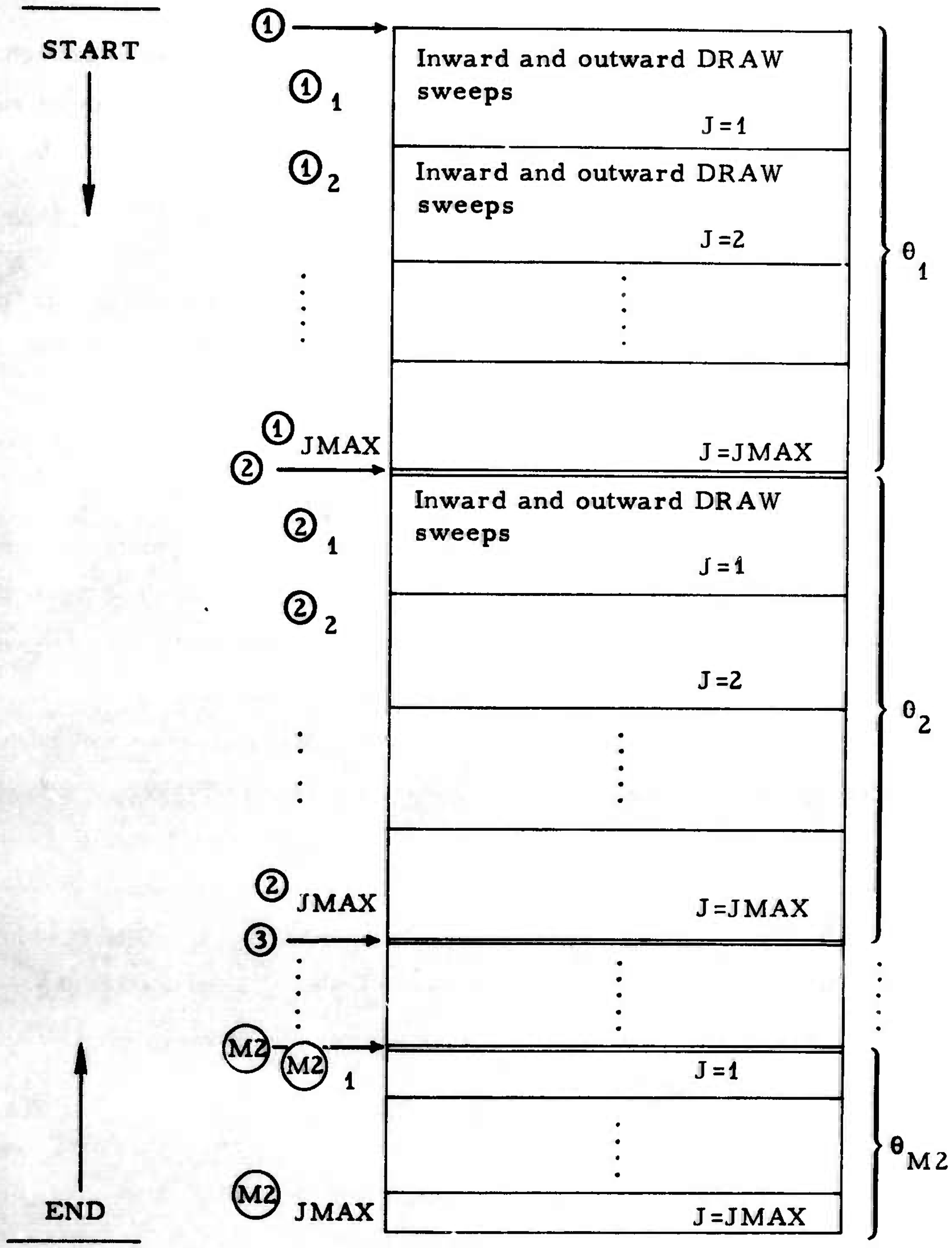
For a given polar angle θ_M transport calculations begin at the system bottom boundary and proceed (or "sweep") upward (if $\theta_M < \pi/2$) or at the system top boundary and sweep downward (if $\theta_M > \pi/2$). Sweeping upward or downward for each zone layer starts at the system outside boundary and proceeds inward, and then outward again until the system outside boundary is reached. For each θ_M, M=1, MUMAX, transport calculations are performed for each zone (I, J) of the entire hydro mesh.

In sweeping $\begin{pmatrix}\text{upward}\\ \text{downward}\end{pmatrix}$, the rate at which energy is leaving the $\begin{pmatrix}\text{top}\\ \text{bottom}\end{pmatrix}$ zone faces of the previous layer of zones in all fans serves as input to the $\begin{pmatrix}\text{bottom}\\ \text{top}\end{pmatrix}$ surface of the current layer of zones. When starting from the system $\begin{pmatrix}\text{bottom}\\ \text{top}\end{pmatrix}$ boundary on a $\begin{pmatrix}\text{upward}\\ \text{downward}\end{pmatrix}$ sweep, the input rates to the $\begin{pmatrix}\text{bottom}\\ \text{top}\end{pmatrix}$ zone faces are calculated from the system $\begin{pmatrix}\text{bottom}\\ \text{top}\end{pmatrix}$ boundary temperatures $\begin{pmatrix}\text{BTHETA(I), I=1, IMAX}\\ \text{ATHETA(I), I=1, IMAX}\end{pmatrix}$, and SUBROUTINES $\begin{pmatrix}\text{BOTBND}\\ \text{TOPBND}\end{pmatrix}$ perform this calculation. When sweeping $\begin{pmatrix}\text{inward}\\ \text{outward}\end{pmatrix}$, the rate at which energy is leaving the $\begin{pmatrix}\text{inside}\\ \text{outside}\end{pmatrix}$ face of the previous zone in all fans L, $\begin{pmatrix}\text{L=1, L2}\\ \text{L=L2P1, LMAX}\end{pmatrix}$, serve as input to the fans LP, $\begin{pmatrix}\text{LP=1, L2}\\ \text{LP=L2P1, LMAX}\end{pmatrix}$, of the $\begin{pmatrix}\text{outside}\\ \text{inside}\end{pmatrix}$ face of the current zone (except

for zones (I,J), J=1,JMAX). When starting an inward sweep through zones (IMAX,J),J=1,JMAX, from the system outside boundary, the input rates to the fans LP, LP=1,L2, are calculated from the system outside boundary temperatures (RTHETA(J),J=1,JMAX) by SUBROUTINE OUTBND.

In performing transportation calculations for a given hydro zone to calculate the rate at which energy is leaving a zone face in a fan L, the ray path of the "characteristic ray" of fan L (See SUBROUTINE DISTNC) is followed assuming that the source is a linear function of path length. The sources at the endpoints of this path are calculated by FUNCTIONS S and SRC using a bilinear interpolation on the zone-centered sources of the current and neighboring zones. If these sources are found to vary significantly (See FUNCTION TRANS) from the zone-centered source of the current zone, the source is assumed to be a constant along the ray path equal to the current zone's zone-centered source.

In sweeping upward or downward, the polar angles θ_M are taken in the order θ_1, θ_{MUMAX}, θ_2, $\theta_{MUMAX-1}$, ..., θ_{M2}, θ_{M2+1} to allow for the possibility of a reflecting system top boundary (a reflecting system bottom boundary cannot be treated with the above programmed sequencing of polar angles). To perform the above sequencing, the "DRAW data" on the "DRAW data" file must be retrieved in a special manner. The "DRAW data" file is composed of blocks of logical records arranged by SUBROUTINE DRAW as shown below.



The input storage buffer array BUFF is filled and used as the transport sweeping occurs, and the "DRAW data" file illustrated above is manipulated in the following manner:

Starting the transport with the "DRAW data" file positioned at point ①,

a. Upward sweep for θ_1: Retrieve data from blocks $①_1$, $①_2$, $①_3$, ..., $①_{JMAX}$ (each contains one or more logical records) as transport sweeping zone layers 1, 2, 3, ..., JMAX respectively.

b. Downward sweep for θ_{MUMAX}: Backspace one block ($①_{JMAX}$) and retrieve data from this block as transport sweeping zone layer JMAX. Then backspace two blocks to the start of block $①_{JMAX-1}$ and retrieve data from this block as transport sweeping zone layer JMAX-1. ... etc.

c. Upward sweep for θ_2: Skip to point $②_{JMAX}$ and retrieve data from blocks $②_1$, $②_2$, $②_3$, ..., $②_{JMAX}$ as transport sweeping zone layers 1, 2, 3, ..., JMAX respectively.

.
.
.

etc.

Note that the number of logical records of data per block $Ⓝ_i$ is a constant for all polar angles θ_M.

For each upward or downward transport sweep associated with polar angle θ_M, the following quantities are updated:

1) ER(K)

2) EBTM, ETOP, ESIDE

3) W2(I)

Note each one of these quantities must be doubled since only one half of the possible fans are considered by DRAW and TRAN2 due to the symmetry inherent in a cylindrical system.

At the end of SUBROUTINE TRAN2 the ER array is checked to see if radiation transport in time DT will change the internal energy of a hydro zone by more than the allowed amount (SLUG); if it will, DT is reduced in TRAN2 to meet this restriction. The internal energy of the

system, ETH, is updated to include energy changes by radiation transport and the variables BACC, SACC, and TACC are updated.

INPUT TO "TRANSPORT SUBROUTINE SEQUENCE"

Results from "DRAW SUBROUTINE SEQUENCE" are transferred to "TRANSPORT SUBROUTINE SEQUENCE."

Conditions that Cause Termination of Execution in "TRANSPORT SUBROUTINE Sequence"

Type	Indicator	Description of Error
INPUT	S1=7.0225	MERGE ≤ 0. and multifrequency problem
INPUT	S1=7.0235	IHNU < 1.
INPUT	S1=7.0250	Q ≥ 0.6 or Q ≤ 0.4.
INPUT	S1=7.0320	XA(K) < 0.
INPUT/ EXECUTION	S1=7.0522	An indefinite number has been created before the call to DVCHK.
INPUT	S1=7.1010	IHNU > NHNU
EXECUTION	S1=7.1068	DT < FFB, the minimum time-step control.

Results from "Transport SUBROUTINE Sequence" Transferred to HECTIC

Medium	Variables
COMMON	(a possible contribution to) DT
	(a contribution to) ETH
	ER(K), K=2,KMAX
	W2(I), I=1,IMAX

Glossary of Variables Used in SUBROUTINE TRAN2

FORTRAN Label	Report Label	Description
AIX(K)	None	The total internal energy of zone K per unit mass (does not include kinetic energy). (in ergs/gm)
AMX(K)	None	The mass of zone K (in gm)
B(K)	None	In the multifrequency calculation, temporary storage
BACC	None	$\int_{t'=0}^{t'=t}$ (rate at which energy is leaving the system bottom boundary due to radiation) dt' (in ergs)
BCATAG	None	Boundary flags. A, B, and R refer to top (above), bottom, and side (right) system boundaries, respectively. If $\theta < 0$, boundary is a perfect reflector; otherwise, it is transmittive to radiation (the side cannot be a reflector).
BCBTAG	None	
BCRTAG	None	
BETA	$\frac{h\nu_1}{\theta}$	In the multifrequency calculation, the lower limit of the current frequency band
BETAP	$\frac{h\nu_2}{\theta}$	In the multifrequency calculation, the upper limit of the current frequency band
BUFF	None	≡ CAP. BUFF(1) ≡ NOI. Input storage array for the "DRAW data" that is to be retrieved from the "DRAW data" file.
C	k_q	C is calculated in the "DRAW subroutine sequence" and used in the transport equation (SUBROUTINE TRANS) in such a way that transport calculations are exactly correct in the case of isotropic, homogeneous radiation with $s = \sigma$ = constant and I_o, the intensity of the radiation field,

		$= \frac{s}{4\pi\sigma}$. (See Equation 177 and FUNCTION TRANS) (in cm^2 steradian) (part of "DRAW data")
D	t_q	Temporary storage for the length of the characteristic ray associated with the current zone and fan (in cm) (part of "DRAW data")
DBGPRT	None	The debug print control: DBGPRT ≥ 10^{-20} means debug print desired; otherwise, debug print not desired
DE	None	In calculating DT in TRAN2, the amount of energy per unit mass which a zone will lose or gain in time DT due to radiation (in ergs/gm)
DFB	None	In the multifrequency calculation, $\frac{15}{\pi^4} \int_{BETAe}^{BETAP} \frac{u^3}{e^u - 1} du$
DHNU	None	In the multifrequency calculation, the width of the current frequency band
DT	None	Time step (which can be controlled by TRAN2) (in sec)
DTEMP	None	Temporary storage
DX(I)	$r_i - r_{i-1}$	Radial zone length of zone (I, J), I=1, JMAX (in cm)
DY(J)	$h=z_j - z_{j-1}$	Vertical zone length of zone (I, J) I=1, IMAX
E	None	The total internal energy per unit mass updated by radiation heating only (in ergs/gm)

EB(I, L)	J_q	Given zone (I, J) and directions specified by M and L: The rate at which radiant energy is entering zone (I, J) through the bottom face (on an upward sweep) or through the top face (on a downward sweep) in fan L in direction MU(M) (in ergs/ sec)
EBTM	None	The rate at which energy is entering the system through the system bottom boundary in time DT due to radiation (in ergs/ sec)
EMB1	None	In the multifrequency calculation, temporary storage
EMB2	None	Same as EMB1
ER(K)	None	The rate at which zone K is gaining energy due to radiation in time DT (in ergs/ sec)
	None	In the multifrequency calculation, temporary storage
ESIDE	None	The rate at which energy is entering the system through the system outside boundary in time DT due to radiation (in ergs/ sec)
ET(I, L)	J_q	Given zone (I, J) and directions specified by M and L: The rate at which radiant energy is leaving zone (I, J) through the top face (on an upward sweep) or through the bottom face (on a downward sweep) in fan L in direction MU(M) (in ergs/ sec)
ETH	None	The total energy in the system at time T (in ergs)

ETOP	None	The rate at which energy is entering the system through the system top boundary in time DT due to radiation (in ergs/sec)
EV(I, L)	J_q	Given zone layer DY(J), zone face X(I), and directions specified by M and L: The rate at which radiant energy is passing through zone surface X(I) in fan L in direction MU(M) (in ergs/sec)
FAC	None	The axial intercept (YA) of the characteristic ray of the current fan in the radiation transport calculation equals Y(J-1/2) + FAC * δy, where δy = YV(L) or YH(L) (See INWARD, OUTWRD, or DISTNC Glossary) In the above expression for YA, FAC = -1., if transporting in an upward direction or = +1., if transporting in a downward direction
FACTOR	None	Temporary storage used in the calculation of ROSS(K) and PLANCK(K)
FFB	None	The minimum time-step allowed for HECTIC (in sec)
FIOUT	None	≡ OLDTH. Saved temporarily on disk in DRAW and restored at the end of TRAN2
FOO	None	The amount of energy gain by the system in time DT due to radiation (in ergs)
GG	None	A dummy argument in the call to ES in TRAN2
HELP	None	Temporary storage

HNU	h_{l}	In the multifrequency calculation; the lower frequency of the current frequency band (in eV)
HNU4	None	= HNU^4
HNUP	h_{i}	In the multifrequency calculation, the upper frequency of the current frequency band (in eV)
HNUP4	None	= $HNUP^4$
I	i	In the radiation transport calculation, the index of the current radial hydro zone with outer radius X(I). Otherwise, a running index
IACT	None	Activity flag for zone K which is used in FUNCTION TRANS: IACT = 1 means zone K is active = 0 means zone K is inactive
IBUFF	None	Index of the last used entry in the current load of the input storage buffer BUFF
IHNU	None	In the multifrequency calculation, the index of the current frequency band
IMAX	i_{max}	The number of radial hydro zones
ISEND	None	Error flag used by SUBROUTINE EDIT
ISWEEP	None	Current direction of TRAN2 sweep: = 1 means sweeping or transporting inward = 2 means sweeping or transporting outward
ITAG	None	Flag for the temperature iteration calculation: = 0 means perform temperature iteration ≠ 0 means do not perform temperature iteration
J	j	In the radiation transport calculation, index of the current axial hydro zone with upper boundary Y(J). Otherwise, a running index
JADD	None	The amount by which the value of J is incremented to give the index of the next layer of zones to be considered in the transport calculation

JMAX	j_{max}	The number of radial hydro zones
K	None	In the radiation transport calculation, the index of the current zone. Otherwise, a running index
K1	None	Temporary storage
K2	None	Temporary storage
KDMY	None	Argument returned by the calls to FUNCTION DVCHK: = 1 means an error = 2 means no error
KFIT	None	An array constructed by HECTIC such that: JMR(KFIT(K), 2) = 1 means zone K is active JMR(KFIT(K), 2) = 0 means zone K is inactive
KMAX	None	KMAX-1 is the total number of hydro zones in the system
L		Fan index
L2	None	LMAX/2
L2P1	None	L2+1
LMAX	ℓ_{max}	One half the total number of fans to be considered in the radiation transport calculation
LP	ℓ'	Fan index
M	m	Index of the current polar angle θ_m (arccos MU(M)) being considered in the radiation transport calculation
M2	None	MUMAX/2
MERGE	None	In the multifrequency calculation, a variable controlling the merging of frequency bands
MFTAG	None	Multifrequency flag: = 0 means monofrequency problem ≠ 0 means multifrequency problem
MMM	None	A running index

MUMAX	2M	The maximum number of discrete polar directions θ_m considered in the radiation transport calculation
N	None	A running index
NB1	None	After an upward transport calculation sweep for a given $M \leq M2$, the downward sweep is performed for direction specified by M' where M' = MUMAX+1-M. NB1 is the number of logical records on the "DRAW data" file that must be skipped (in a backward direction) so that inward and outward transport sweeping may be performed on the zone layer JMAX
NBA	None	After sweeping inward and outward one zone layer in transporting in a downward direction, NBA = the number of records that must be skipped (in a backward direction) on the "DRAW data" file before inward and outward transport sweeping may begin on the next lower zone layer
NC	None	Integer value of cycle number
NHNU	None	In the multifrequency calculation, the total number of frequency bands
NOBUFF	None	The number of input buffer loads BUFF that will be filled for an inward or outward transport sweep along one horizontal layer of zones (J) in direction specified by M (not a function of M or J)
NOI	None	BUFF(1) The number of inward or outward DRAW zone sweeps contained in the current input buffer BUFF
NSKP	None	After sweeping in a downward direction M(>M2), the number of logical records of the "DRAW data" file that must be skipped in a forward direction before an upward sweep in direction specified by M' = MUMAX-M+2 can begin

NVEZ	None	In the temperature iteration calculation, the temperature iteration counter
NY	None	Not used
OLDTH	None	≡ FIOUT
P	None	≡ PLANCK. Saved temporarily on disk in DRAW and restored at the end of TRAN2
PI	π	π
PLANCK(K)	None	The Planck mean opacity of zone K across the merged frequency band (in $\frac{1}{cm}$)
PUR	None	Not used
PUZ	None	Not used
Q	None	In the calculation of DT in TRAN2, the amount by which the internal energy per unit mass of a zone is allowed to change due to radiation transport (in ergs/gm) Also, in the multifrequency calculation, a parameter used in the merging of frequency bands
ROSS(K)	σ	The Rosseland mean absorption coefficient for zone K (in 1/cm)
RPTAG	None	Absorption coefficient flag: ≠ 0 means set the Planck mean absorption coefficient equal to the Rosseland mean absorption coefficient = 0 means keep both absorption coefficients distinct
RUR	None	Not used
RUZ	None	Not used
S1	None	Error flag for EDIT

SACC	None	$\int_{t'=0}^{t'=t}$ (rate at which energy is leaving the system bottom boundary due to radiation) dt' (in ergs)
SLUG	None	In the calculation of DT in TRAN2, the fraction by which the internal energy per unit mass of a zone is allowed to change due to radiation transport
SP	$4\pi S^c$	The source value (the rate at which radiant energy is being generated per unit volume) assigned to the midpoint of zone K having temperature THETA(K) and absorption coefficient ROSS(K) $= 4 * \text{ROSS(K)} * \text{STEF} * \text{THETA}^4\text{(K)}$ (in ergs/cm^3)
STEF	$\frac{ac}{4}$	Stefan's constant = 1.0283×10^{12} (in $\frac{\text{ergs}}{\text{cm}^2 \text{ sec eV}^4}$)
SUM	None	In the radiation transport calculation across a given zone (I,J) in a direction specified by fan L and direction M: $= \sum_{LP=1,LMAX}(\text{appropriate } \eta) \cdot (\text{appropriate EV})$ $+ \sum_{LP=1,LMAX}(\text{appropriate } \eta) \cdot (\text{appropriate EB})$ $= \begin{pmatrix} \text{SUMO} \\ \text{or} \\ \text{SUMI} \end{pmatrix} + \text{SUMB}$ (in ergs/sec)
SUMB	None	See SUM
SUMO	None	See SUM
SV	None	Temporary storage
T4	None	In merging frequency bands, $= \text{THETA(K)}^4$ (in eV^4)

TACC	None	$\int_{t'=0}^{t'=t}$ (rate at which energy is leaving the system top boundary due to radiation) dt' (in ergs)
TAU(I)	$\pi(r_i^2 - r_{i-1}^2)$	The area of the annulus defined by radial hydro interval I (in cm^2)
TEMP(1)	None	In changing DT in TRAN2, the computed fraction by which DT is reduced to comply with the restrictive variable SLUG
THETA(K)	θ	Temperature of zone K (in eV)
THTAMX	None	In the multifrequency calculation, the highest temperature in the system used in deciding whether to merge frequency bands (in eV)
U	None	≡ ROSS. Saved temporarily on disk in DRAW and restored at the end of TRAN2
V	None	≡ XA ≡ EB. Saved temporarily on disk in DRAW and restored at the end of TRAN2
VEZ	None	Not used
W2(I)	None	The rate at which energy is leaving the system top boundary adjacent to zone (I, JMAX) due to radiation, divided by TAU(I) (in $\frac{ergs}{cm^2 sec}$)
X(I)	r_i	Outer radius of radial hydro zone I
XA	None	Temporary storage for the radial coordinate of the intersection of the current fan's characteristic ray and the "nearest zone surface." (in cm) (part of "DRAW data")

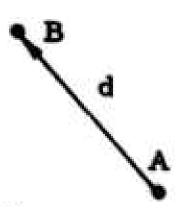
XHH	None	Radial coordinate of the midpoint of the current zone, $X(I-\frac{1}{2})+DX(I)/2$ (in cm)
Y(J)	z_j	The vertical coordinate of the upper boundary of hydro zones (I, J), I=1, IMAX (in cm)
YA	None	Temporary storage for the axial coordinate of the intersection of the current fan's characteristic ray and the "nearest zone surface" (in cm) (part of "DRAW data")
YB	z_j or z_{j-1}	Temporary storage (=Y(J) or Y(J-1))
YH2	None	Y(J)-DY(J)/2, the axial coordinate of the midpoint of zones (I, J), I=1, IMAX (in cm)

FLOW OF CONTROL IN FUNCTION TRANS (SIGMA, D, XA, XB, YA, YB, C, SUM, IACT, SP)

FUNCTION TRANS performs a transport calculation across the current zone of SUBROUTINE TRAN2 yielding the rate at which energy is being transported through face S of the current zone in fan L by the equation:

$$E_{S,L} = \left(\sum_{\substack{S',LP \\ S' \neq S}} \eta_{\substack{(L,LP) \\ S' \to S}} \cdot E_{S',LP} \right) * e^{-\sigma d}$$

$$+ C \left\{ \frac{S_A}{4\pi\sigma} \left[1 - e^{-\sigma d} \right] + \frac{S_B - S_A}{4\pi\sigma^2} \left[\frac{\sigma d - 1 + e^{-\sigma d}}{d} \right] \right\}$$



where

σ = optical absorption coefficient associated with the current zone

d = length in the current zone of the characteristic ray of the current fan L.

S_A = the interpolated source at coordinate A, the starting point of the transport along d.

S_B = the interpolated source at coordinate B, the end point of the transport along d.

$\eta(L,LP)_{S'\to S}$ = "view factor" that couples the rate at which energy is entering the current zone through face S' in fan LP to the rate at which energy is leaving the current zone through face S in fan L. (See SUBROUTINES DRAW, INWARD, and OUTWRD.)

$E_{P',LP}$ = the rate at which energy is entering the current zone through face S' in fan LP.

$\sum_{\substack{S',LP \\ S'\neq S}} \eta(L,LP)_{S'\to S} \, E_{S',LP}$ = the rate at which energy is leaving the current zone through face S in fan L assuming $\sigma = S_A = S_B = 0$.

C = A constant calculated such that if $S=\sigma=$ a constant for the current zone then the transport calculation will be exact

S_A and S_B are obtained from a bilinear interpolation of the zone-centered sources of the current zone and its neighboring zones. If these interpolated values S_A and S_B and the zone-centered source S_P of the current zone differ by an amount FAC, the S_A and S_B are set equal to S_P. The source is assumed to vary linearly along the path from A to B for the transport calculation.

Glossary of Variables Used in FUNCTION TRANS (SIGMA, D, XA, XB, YA, YB, C, SUM, IACT, SP)

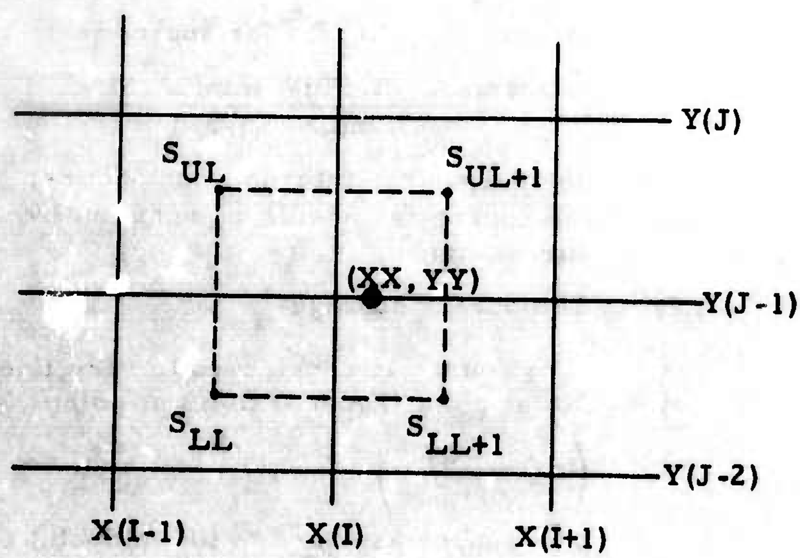
FORTRAN Label	Report Label	Description
A1	None	Temporary storage
A2	None	Temporary storage
A3	None	Temporary storage

A4	None	Temporary storage
A5	None	Temporary storage
C	k_q	Constant calculated such that the transport equation will yield an exact solution in the case of isotropic, homogeneous radiation with $s=\sigma=$ constant and I_o, the intensity of the radiation field, $= \frac{s}{4\pi\sigma}$ (in cm^2 steradian)
D	t_q	The length of the characteristic ray of the current fan in the current zone of SUBROUTINE TRAN2 (in cm)
FAC	f	Magnitude criterion for source term interpolation: if $\frac{\max(SA, SB, SP) - \min(SA, SB, SP)}{\min(SA, SB, SP)} < FAC$, normal source interpolation occurs; otherwise, the source is considered constant over the current zone
HELP	None	Temporary storage
SA	$4\pi S_i^i$	The source is considered to vary linearly from SA at point (XA, YA) to SB at point (XB, YB) $\left(\text{in } \frac{\text{ergs}}{\text{cm}^3\text{sec}}\right)$
SB	$4\pi S^o$	
SP	$4\pi S^c$	The source assigned to the midpoint of the current zone which has absorption coefficient SIGMA $\left(\text{in } \frac{\text{ergs}}{\text{cm}^3\text{ sec}}\right)$
SIGMA		The Rosseland mean absorption coefficient for the current zone. (in $\frac{1}{cm}$)
SUM	None	In the radiation transport calculation across the current zone: $= \sum_{LP=1, LMAX}$ (appropriate η) · (appropriate EV)

		$+ \sum_{LP=1, LMAX}$ (appropriate η)·(appropriate EB)
		(in ergs/sec
		(see TRAN2 glossary)
XA, XB, YA, YB	None	The radiation transport calculation is performed along the characteristic ray path from point (XA, YA) to point (XB, YB)
		Note: (XB, YB)-(XA, YA) = D
		(in cm)

FLOW OF CONTROL IN FUNCTION S(XX, YY)

Given coordinates (XX, YY), FUNCTION S calculates the source at this point with a bilinear interpolation on the zone-centered sources of the neighboring zones.



If point (XX, YY) is such that

$$X(1) - \frac{DX(1)}{2} \leq XX \leq X(IMAX) - \frac{DX(IMAX)}{2}$$

and

$$Y(1) - \frac{DY(1)}{2} \leq YY \leq Y(JMAX) - \frac{DY(JMAX)}{2}$$

an imaginary "source zone" is created that contains the point (XX, YY) and has associated with its vertices the zone-centered sources (S_{UL}, S_{UL+1}, S_{LL}, S_{LL+1}) of the neighboring zones which are obtained from

FUNCTION SRC. A bilinear interpolation is performed on the vertices of the "source zone" to obtain the source at point (XX, YY).

If point (XX, YY) lies outside the above limits, interpolation for the source at this point is performed according to the following algorithm:

a. If $XX > X(IMAX) - \frac{DX(IMAX)}{2}$ and $[YY < Y(JMAX) - \frac{DY(JMAX)}{2}$ or $YY > Y(1) - \frac{DY(1)}{2}]$, a linear interpolation is performed between S_{UL} and S_{LL} on YY.

b. If $XX < X(1) - \frac{DX(1)}{2}$ and $[YY < Y(JMAX) - \frac{DY(JMAX)}{2}$ or $YY > Y(1) - \frac{DY(1)}{2}]$, a linear interpolation is performed between S_{UL+1} and S_{LL+1} on YY.

c. If $YY > Y(JMAX) - \frac{DY(JMAX)}{2}$ and $[XX < X(IMAX) - \frac{DX(IMAX)}{2}$ or $XX > X(1) - \frac{DX(1)}{2}]$, a linear interpolation is performed between S_{LL} and S_{LL+1} on XX.

d. If $YY < Y(1) - \frac{DY(1)}{2}$ and $[XX < X(IMAX) - \frac{DX(IMAX)}{2}$ or $XX < X(1) - \frac{DX(1)}{2}]$, a linear interpolation is performed between S_{UL} and S_{UL+1} on XX.

e. Otherwise, the source at (XX, YY) is set equal to the nearest zone-centered source.

Glossary of Variables Used in FUNCTION S (XX, YY)

FORTRAN Label	Report Label	Description
ALL	None	Temporary storage
ALR	None	Temporary storage

AUL	None	Temporary storage
AUR	None	Temporary storage
DELT	None	DXP * DYP (in cm^2)
DX(I)	None	Radial zone length of hydro zones (I, J), J=1, JMAX (in cm)
DXP	None	Radial "source zone" length (in cm)
DY(J)	h	Vertical length of hydro zones (I, J), J=1, JMAX (in cm)
DYP	None	Vertical "source zone" length (in cm)
I	i	Index of current radial hydro zone with outer radius X(I)
ILFT		The inner face of the "source zone" is contained within the vertical layer of zones (ILFT, J), J=1, JMAX
IMAX	i_{max}	The number of radial hydro zones
J	j	Index of the current axial hydro zone with upper boundary Y(J)
JLOW	None	The lower face of the "source zone" is contained within the horizontal layer of zones (I, JLOW), I=1, IMAX
JMAX	j_{max}	The number of axial hydro zones
KLL	None	Index of the hydro zone containing the lower left corner of the "source zone"
KUL	None	Index of the hydro zone containing the upper left corner of the "source zone"
X(I)	r_i	Outer radius of radial hydro zone I (in cm)
XLFT	None	Inner radius of the "source zone" (in cm)

XRGT	None	Outer radius of the "source zone" (in cm)
XX	None	The radius at which the source term is to be calculated (in cm)
Y(J)	z_j	The vertical coordinate of the upper boundary of hydro zones (I, J), I=1, IMAX (in cm)
YDN	None	Axial coordinate of lower face of the "source zone" (in cm)
YUP	None	Axial coordinate of the upper face of the "source zone" (in cm)
YY	None	The axial coordinate at which the source term is to be calculated (in cm)

FLOW OF CONTROL IN FUNCTION SRC (KKK)

Given hydro zone KKK, the FUNCTION SRC calculates the source associated with the midpoint of the zone. If the zone is considered "inactive" by HECTIC, then the source is zero; otherwise,

$$S_{KKK} = \sigma_{KKK}\, ac\, \theta^4_{KKK}$$

or

$$S_{KKK} = \sigma_{KKK}\, ac\, \theta^4_{KKK} \int_{\frac{h\nu_1}{\theta_{KKK}}}^{\frac{h\nu_2}{\theta_{KKK}}} \frac{15}{\pi^4} \frac{u^3 du}{e^u - 1}$$

if the problem is monofrequency or multifrequency, respectively, where

$\frac{ac}{4}$ = Stefan's constant

ν_1, ν_2 = the frequency limits

h = Planck's constant

σ_{KKK} = Rosseland mean absorption coefficient of zone (KKK)

θ_{KKK} = Temperature of zone (KKK)

S_{KKK} = zone-centered source for zone (KKK)

Glossary of Variables Used in FUNCTION SRC (KKK)

FORTRAN Label	Report Label	Description
DFB	None	In the multifrequency calculation, $\frac{15}{\pi^4}\int_A^B \frac{u^3}{e^u-1}\,du$
HNU	$h\nu_1$	In the multifrequency calculation, the lower frequency of the current frequency band (in eV)
HNUP	$h\nu_2$	In the multifrequency calculation, the upper frequency of the current frequency band (in eV)
KFIT	None	See SUBROUTINE TRAN2
KKK	None	Index of the zone in which tne source term is to be calculated
MFTAG	None	Multifrequency flag: = 0 means monofrequency frequency problem; ≠ 0 means multifrequency problem
ROSS(KKK)	σ	The Rosseland mean absorption coefficient for zone KKK (in $\frac{1}{cm}$)

STEF	$\frac{ac}{4}$	Stefan's constant = 1.0283×10^{12} (in $\frac{\text{ergs}}{\text{cm}^2 \text{ sec eV}^4}$)
THETA(KKK)	θ	Temperature of zone KKK (in eV)

FLOW OF CONTROL IN SUBROUTINE BOTBND

For a given polar angle θ_m(arccos MU(M)) and a system bottom boundary temperature array B_θ (BTHETA(I), I=1,IMAX), SUBROUTINE BOTBND calculates the rate at which energy is entering each zone of the layer of zones (I,1), I=1,IMAX, in each fan L,L=1,LMAX, for a monofrequency HECTIC problem.

The system bottom boundary is considered to be a black-body surface divided into IMAX annuli, each annulus I having area π $(X(I)^2 - X(I-1)^2)$ and an associated black-body temperature BTHETA(I). The intensity of radiation, BINT, emitted into the system through each annulus I of this black-body surface integrated over all frequencies is given by

$$\text{BINT} = \frac{ac}{4\pi} B_\theta^{\,4}$$

where $\frac{ac}{4}$ = Stefan's constant

The rate at which energy is entering zone (I,1) which borders annulus I in fan L is given by

EB(I,L) = (BINT) * (the rate at which energy is entering zone (I,1) through its bottom face in fan L in the case of isotropic, homogeneous radiation σ=S=0 with the intensity of the radiation field = 1)

Glossary of Variables Used in SUBROUTINE BOTBND

FORTRAN Label	Report Label	Description
BINT	None	Temporary storage
BTHETA(I)	θ_{out}	The temperature of the black-body system boundary adjacent to zone (I, 1) (in eV)
EB(I, L)	J_q	Given zone (I, J) and directions specified by M and L: The rate at which radiant energy is entering zone (I, J) through its bottom face in fan L and direction MU(M) (in ergs/sec)
I	i	A running index referring to the radial hydro zones with outer radius X(I)
IMAX	i_{max}	The number of radial hydro zones
JB(1, I, L, M)	k_q	The FUNCTION JB returns a value which is the rate at which energy is entering zone (I, 1) through the face Y(0) in fan L in direction MU(M) in the case of isotropic, homogeneous radiation (σ=S=0) with I_o, the intensity of the radiation field, = 1. (in ergs/sec)
L	ℓ	A running index referring to the fan L
LMAX	ℓ_{max}	The total number of fans to be considered
M	m	Index of the current polar angle θ_m(arccos MU(M)) being considered
PI	π	π
STEF	$\frac{ac}{4}$	Stefan's constant = 1.0283 x 10^{12} $\left(\text{in } \frac{\text{ergs}}{\text{cm}^2 \text{ sec eV}^4}\right)$

FLOW OF CONTROL IN SUBROUTINE OUTBND

For a given polar angle θ_m (arccos MU(M)) and a particular section DY(J) of the system outside boundary with boundary temperature array R_θ (RTHETA(J), J=1,JMAX), SUBROUTINE OUTBND calculates the rate at which energy is entering zone (IMAX, J) in each fan L, L=1,L2, for a monofrequency HECTIC problem.

The system outside boundary is considered to be a black-body surface divided into JMAX surfaces, each surface DY(J) having area 2*π*X(IMAX)*DY(J) and an associated black-body temperature RTHETA(J). The intensity of radiation, BINT, emitted into the system through surface DY(J) of this black-body surface integrated over all frequencies is given by

$$\text{BINT} = \frac{ac}{4\pi} R_\theta^4$$

where

$$\frac{ac}{4} = \text{Stefan's constant}$$

The rate at which energy is entering in fan L of zone (IMAX, J) with outside face DY(J) is given by

EV(I, L) = (BINT)* (the rate at which energy is entering zone (IMAX, J) through its outside face in fan L in the case of isotropic, homogeneous radiation σ=S=0 with the intensity of the radiation field = 1).

Glossary of Variables Used in SUBROUTINE OUTBND

FORTRAN Label	Report Label	Description
BINT	None	Temporary storage
EV(IMAX, L)	J_q	Given zone layer DY(J), surface X(IMAX), and directions specified by M and L: The rate at which radiant energy is entering surface DY(J) in fan L in direction MU(M) (in ergs/sec)

IMAX	i_{max}	The number of radial hydro zones
J	j	Index of the current layer of hydro zones DY(J)
JO(J, IMAX, L, M)	k_q	The FUNCTION JO returns a value which is the rate at which energy is entering zone (IMAX, J) through face X(IMAX) in fan L in direction MU(M) in the case of isotropic, homogeneous radiation σ=S=0 with I_o, the intensity of the radiation field, = 1 (in ergs/sec)
L	ℓ	A running index referring to the fan L
L2	None	LMAX/2
M	m	Index of the current polar angle θ_m (arccos MU(M)) being considered
PI	π	π
RTHETA(J)	θ_{out}	The temperature of the black-body system boundary adjacent to zone (IMAX, J) (in eV)
STEF	$\frac{ac}{4}$	Stefan's constant = 1.0283×10^{12} $\left(\text{in } \frac{\text{ergs}}{\text{cm}^2 \text{ sec eV}^4}\right)$

FLOW OF CONTROL IN SUBROUTINE TOPBND

For a given polar angle θ_m (arccos MU(M)) and a top (above) system boundary temperature array A_θ (ATHETA(I), I=1, IMAX), SUBROUTINE TOPBND calculates the rate at which energy is entering each zone of the layer of zones (I, JMAX), I=1, IMAX, in each fan L, L=1, LMAX for a monofrequency HECTIC problem.

The system top boundary is considered to be a black-body surface divided into IMAX annuli, each annulus I have area $\pi(XI)^2$-$X(I-1)^2$) and an associated black-body temperature ATHETA(I). The intensity of radiation,

BINT, emitted into the system through each annulus I of this black-body surface integrated over all frequencies is given by:

$$\text{BINT} = \frac{ac}{4\pi} A_{\theta}^{4}$$

where

$$\frac{ac}{4} = \text{Stefan's constant}$$

The rate at which energy is entering zone (I, JMAX) in fan ℓ through annulus I is given by

EB(I, L) = (BINT) * (the rate at which energy is entering zone (I, JMAX) through its top face in fan L in the case of isotropic, homogeneous radiation σ =S=0 with the intensity of the radiation field = 1.)

Glossary of Variables Used in SUBROUTINE TOPBND

FORTRAN Label	Report Label	Description
ATHETA(I)	θ_{out}	Temperature of the black-body system boundary adjacent to zone (I, JMAX) (in eV)
BCATAG	None	Boundary flag referring to the above (top) system boundary < 0 means boundary is a perfect reflector ≥ 0 means boundary is transmittive to radiation
BINT	None	Temporary storage
EB(I, L)	J_q	Given zone (I, J) and directions specified by M and L: The rate at which radiant energy is entering zone (I, J) through its top face in fan L in direction MU(M) (in ergs/sec)
I	i	A running index referring to the radial hydro zones with outer radius X(I)
IMAX	i_{max}	The number of radial hydro zones

JB(JMAX,I, L,M)	k_q	The FUNCTION JB returns a value which is the rate at which energy is entering zone (I,JMAX) through the face Y(JMAX) in fan L in direction MU(M) in the case of isotropic, homogeneous radiation σ=S=0 with I_o, the intensity of the radiation field, = 1 (in ergs/sec)
IMAX	i_{max}	The number of axial hydro zones
L	ℓ	A running index referring to fan L
LMAX	ℓ_{max}	The total number of fans to be considered given θ_m
M	m	The index of the current polar angle θ_m (arccos MU(M)) being considered
PI	π	π
STEF	$\frac{ac}{4}$	Stefan's constant = 1.0283×10^{12} $\left(\text{in } \frac{\text{ergs}}{\text{cm}^2\,\text{sec eV}^4}\right)$

SECTION VIII

NONEQUILIBRIUM DIFFUSION IN HECTIC

STATEMENT OF THE PROBLEM

The basis for the nonequilibrium diffusion approximation (see Section IV of Ref. 4 and Ref. 8) is the assumption that the second and higher directional moments of the radiant intensity are negligible, so that

$$I(\vec{\Omega}) = \frac{c\ E + 3\vec{F}\cdot\vec{\Omega}}{4\pi} \tag{181}$$

where E is the density and $\vec{F}$ is the flux of radiant energy. When this assumption is made, Eq. (102) can be written as two equations (Eq. (52) of Ref. 4):

$$\begin{aligned} \nabla\cdot\vec{F} + c\ \sigma\ E &= s \\ \frac{c}{3}\ \nabla E + \sigma\ \vec{F} &= 0 \end{aligned} \tag{182}$$

where

$$s = \int S\ d\ \Omega \tag{183}$$

in which the direction variable $\vec{\Omega}$ no longer appears. The flux $\vec{F}$ may also be eliminated, leaving just one second-order equation

$$-\frac{c}{3\sigma}\ \nabla^2\ E + c\ \sigma\ E = s \tag{184}$$

for the density E.

The program TDRAD is a subroutine designed to be used in HECTIC for calculating radiative transfer using Eq. (184). To obtain numerical estimates of the energy density E, TDRAD solves the following set of finite-difference-equations (Eq. (56) of Ref. 4):

$$B^L_{i,j}(E_{i,j} - E_{i-1,j}) + B^R_{i,j}(E_{i,j} - E_{i+1,j}) + B^A_{i,j}(E_{i,j} - E_{i,j+1}) + B^B_{i,j}(E_{i,j} - E_{i,j-1}) + A_{i,j}E_{i,j} = D_{i,j} \tag{185}$$

where $E_{i,j}$ is the density of radiant energy in zone $Z_{i,j}$ and

$$\begin{aligned} B^L_{i,j} &= \frac{4r_{i-1}}{3(r_i^2 - r_{i-1}^2)\left[\sigma_{i,j}(r_i - r_{i-1}) + \sigma_{i-1,j}(r_{i-1} - r_{i-2})\right]} \\ B^R_{i,j} &= \frac{4r_i}{3(r_i^2 - r_{i-1}^2)\left[\sigma_{i,j}(r_i - r_{i-1}) + \sigma_{i+1,j}(r_{i+1} - r_i)\right]} \\ B^A_{i,j} &= \frac{2}{3(z_j - z_{j-1})\left[\sigma_{i,j}(z_j - z_{j-1}) + \sigma_{i,j+1}(z_{j+1} - z_j)\right]} \\ B^B_{i,j} &= \frac{2}{3(z_j - z_{j-1})\left[\sigma_{i,j}(z_i - z_{j-1}) + \sigma_{i,j-1}(z_{i-1} - z_{j-2})\right]} \end{aligned} \tag{186}$$

$$A_{i,j} = \sigma_{i,j}$$

$$D_{i,j} = S_{i,j}$$

Let k be a single index that enumerates zones. That is, let

$$k = (j - 1)i_{max} + i$$

(Traditionally, the FORTRAN subscript K for zone $Z_{i,j}$ is k + 1 in OIL and its derivatives). Then, Eq. (185) can be conveniently written

$$\sum_{m'} B_{k,m} x_m = s_k \tag{187}$$

or, in matrix notation,

$$B x = s \tag{188}$$

where

$$\left.\begin{aligned}
x_k &= E_{i,j} \\
s_k &= D_k \\
B_{k,k-1} &= -B^L_{i,j} \\
B_{k,k+1} &= -B^R_{i,j} \\
B_{k,k+I} &= -B^A_{i,j} \\
B_{k,k-I} &= -B^B_{i,j} \\
B_{k,k} &= A_{i,j} + B^L_{i,j} + B^R_{i,j} + B^A_{i,j} + B^B_{i,j} \\
B_{k,m} &= 0 \quad \text{otherwise}
\end{aligned}\right\} \tag{189}$$

where $I = i_{max}$.

A method of successive approximations due to Oliphant (Ref. 9) is used in TDRAD to solve Eq. (188). It was soon discovered that the rate of convergence of Oliphant's method is often very low, especially when the zones of the HECTIC mesh are optically thin. Intensive effort has been exerted in the search for efficient methods of solutions of systems of the form of Eq. (185), especially in the fields of thermal and neutron diffusion (Ref. 10).

The most successful procedures have been successive over-relaxation, or SOR, and alternating-direction-implicit, or ADI. Accordingly, tests were conducted to compare the performance of these procedures with Oliphant's to see which would be most suitable for application to problems of the type that TDRAD is designed to treat.

FORTRAN expressions for the above-mentioned variables are listed below, where

$$K = k + 1 = (j - 1)\, i_{max} + i + 1$$

$$B^L_{i,j} = XA(K)$$

$$B^R_{i,j} = XC(K)$$

$$B^B_{i,j} = XALP(K)$$

$$B^A_{i,j} = XGAM(K)$$

$$D_{i,j} = XD(K)$$

$$A_{i,j} = -XA(K)-XB(K)-XC(K)-XALP(K)-XGAM(K)$$

$$E_{i,j} = ERAD(K)$$

The appendix contains listings of the FORTRAN source statements for a version of TDRAD that uses the ADI method. The programming of Oliphant's method and SOR is included in supplementary listings.

COMMON FEATURES OF THE METHODS

All three methods investigated are based on the idea of using an approximate inverse of the matrix B to obtain a correction to an estimate of the solution of Eq. (188). The methods differ mainly in their procedures for obtaining the approximate inverse.

To be more explicit, suppose C is an approximate inverse of B in the

sense that

$$CB = I - E \tag{190}$$

where I is the identity and E is small. Suppose further that x^{ℓ} is the ℓ^{th} in a series of estimates of the solution of $Bx = s$.

The correction to x^{ℓ} is taken to be $C(s - Bx^{\ell})$, so the $\ell + 1^{st}$ estimate becomes

$$x^{\ell+1} = x^{\ell} + C(s - Bx^{\ell}) \tag{191}$$

Let $By = s$ and let the error in x^{ℓ} be $e^{\ell} = y - x^{\ell}$. Then $s - Bx^{\ell} = s - B(y - e^{\ell}) = Be^{\ell}$, so

$$\begin{aligned} e^{\ell+1} &= y - x^{\ell+1} \\ &= y - \left[x^{\ell} + C(s - Bx^{\ell})\right] \\ &= y - (x^{\ell} + CBe^{\ell}) \\ &= e^{\ell} - CBe^{\ell} \\ &= (I - CB)\, e^{\ell} \\ &= Ee^{\ell} \end{aligned}$$

It follows that $e^{\ell} = E^{\ell} e^{o}$, where $E^{o} = I$ and $E^{\ell+1} = EE^{\ell}$, the $\ell+1^{st}$ power of E. Therefore, $x^{\ell} \to y$ provided that E is so small that the moduli of its eigenvalues are all less than unity.

Another common feature of the various methods is the use of convergence accelerating parameters. What is done amounts to setting

$$x^{\ell+1} = x^{\ell} + \omega C(s - Bx^{\ell}) \tag{192}$$

where ω is an adjustable scalar parameter. It is often true that methods of successive approximation of the type described by Eq. (191) underestimate the size of the correction, and using Eq. (192) with $\omega > 1$, but not too large, will improve the convergence rate. Actually, Eq. (192) is not used exactly as written in any of the three methods that were tested, as will be seen in the following detailed discussion of the methods.

OLIPHANT'S METHOD

Oliphant's method (Ref. 9) is, briefly, to modify the matrix B in such a way that it becomes easy to invert with the expectation that if the modification is not too great, then the inverse of the new matrix will closely approximate that of B.

First, however, let U be the upper part of B: i.e., let

$$U_{k,m} = \begin{cases} B_{k,m} & \text{if } m > k \\ 0 & \text{otherwise} \end{cases} \tag{193}$$

and set

$$C = B + \lambda U \tag{194}$$

The next step is to find a small matrix H such that C + H may be expressed as a product of easily determined triangular factors. Thus, let

$$C + H = WV \tag{195}$$

where

$$\begin{aligned} W_{k,\ell} &= 0 \qquad \text{for } \ell > k \\ V_{\ell,m} &= 0 \qquad \text{for } \ell > m \end{aligned} \tag{196}$$

The particular solution of Eq. (195) that is used is

$$\begin{aligned} W_{k,k-I} &= B_{k,k-I} \\ W_{k,k-1} &= B_{k,k-1} \\ W_{k,k} &= B_{k,k} - W_{k,k-I} V_{k-I,k} - W_{k,k-1} V_{k-1,k} \\ W_{k,m} &= 0 \quad \text{for } m = k,\ k-1, \text{ or } k-I \end{aligned}$$

$$
\begin{aligned}
V_{k,k+I} &= (\lambda + 1)\ B_{k,k+I}/W_{k,k} \\
V_{k,k+1} &= (\lambda + 1)\ B_{k,k+1}/W_{k,k} \qquad (197) \\
V_{k,k} &= 1 \\
V_{k,m} &= 0 \quad \text{for } m \neq k,\ k+1 \text{ or } k+I \\
H_{k,k+I-1} &= W_{k,k-1}\ V_{k-1,\ k+I-1} \\
H_{k,k-I+1} &= W_{k,k-I}\ V_{k-I,\ k-I+1} \\
H_{k,m} &= 0 \quad \text{for } m \neq k - I+1 \text{ or } k + I - 1
\end{aligned}
$$

where $I = i_{max}$.

Note that $W_{k,m}$ and $V_{k,m}$ are expressed in terms of B and $W_{k',m'}$ and $V_{k',m'}$ for $k' < k$ or $k' = k$ and $m' < m$, so that there is no problem of finding a sequence in which to carry out the indicated operations. The verification of Eq. (197) is accomplished by direct substitution in Eq. (195). Adding $(\lambda U + H)\ x$ to both sides of Eq. (188) yields

$$(B + \lambda\ U + H)\ x = s + (\lambda U + H)\ x$$

or

$$W\ V\ x = s + (\lambda U + H)\ x \qquad (198)$$

so that

$$x = V^{-1}W^{-1}\left[s + (\lambda U + H)x\right] \qquad (199)$$

If x^{ℓ} is a trial solution, one revises it by substituting it for x in the RHS of Eq. (199). Thus,

$$x^{\ell+1} = V^{-1}W^{-1}\left[s + (\ U + H\ \ x^{\ell}\right] \qquad (200)$$

The matrix $V^{-1}W^{-1}$ plays the role of approximate inverse of B. To see the analogy between Eq. (191) and Eq. (200), observe that

$$x^{\ell+1} = x^{\ell} + V^{-1}W^{-1}\left[s + (\lambda U + H)\, x^{\ell} - W\, V\, x^{\ell}\right]$$

$$= x^{\ell} + V^{-1}W^{-1}\,(s - Bx^{\ell})$$

The numerical work is rather simple. To see what is involved, rewrite Eq. (199) in three parts as follows:

$$\left.\begin{aligned} y &= s + (\lambda U + H)\, x^{\ell} \\ Wz &= y \\ V\, x^{\ell+1} &= z \end{aligned}\right\} \qquad (201)$$

Writing these equations out in full yields

$$\begin{aligned} y_k = s_k &+ \lambda B_{k,k+1}\, x^{\ell}_{k+1} + \lambda B_{k,k+I}\, x^{\ell}_{k+I} \\ &+ H_{k,k+I-1}\, x^{\ell}_{k+I-1} + H_{k,k-I+1}\, x^{\ell}_{k-I+1} \end{aligned} \qquad (202)$$

$$z_k = (y_k - W_{k,k-1}\, z_{k-1} - W_{k,k-I}\, z_{k-I})/W_{k,k} \qquad (203)$$

$$x^{\ell+1}_k = z_k - V_{k,k+1}\, x^{\ell+1}_{k+1} - V_{k,k+I}\, x^{\ell+1}_{k+I} \qquad (204)$$

Equations (202) and (203) are evaluated simultaneously in order of increasing k, so the y's need not actually be stored. Equation (205) is then evaluated in reverse order so that $x^{\ell+1}_{k+1}$ and $x^{\ell+1}_{k+I}$ will be available for calculating $x^{\ell+1}_k$. Oliphant's acceleration procedure amounts to replacing Eq. (204) by

$$x^{\ell+1}_k = \omega(z_k - V_{k,k+1}\, x^{\ell+1}_{k+1} - V_{k,k+1}\, x^{\ell+1}_{k+1}) + (1 - \omega)\, x^{\ell}_k \qquad (205)$$

Usually, $-1 \le \lambda$ 1 and $0.5 \le \omega \le 1.5$.

FORTRAN expressions for the above-mentioned variables are supplied in the following list (because of the transience of FORTRAN assignments, "x = A and y = A" does not imply "x = y"):

$$k = K-1$$

$$I = IMAX$$

$$B_{k,k} = XB(K)$$

$$B_{k,k-1} = XA(K)$$

$$B_{k,k+1} = XC(K)$$

$$B_{k,k-I} = XALP(K)$$

$$B_{k,k+I} = XGAM(K)$$

$$x_k^{\ell} = ERAD(K)$$

$$s_k = -XD(K)$$

$$\lambda = CMXK - 1.0$$

$$U_{k,k+1} = XC(K)$$

$$U_{k,k+I} = XGAM(K)$$

$$W_{k,k} = XB(K)$$

$$W_{k,k-1} = XA(K)$$

$$W_{k,k-I} = XALP(K)$$

$$V_{k,k+I} = XGAM(K)$$

$$V_{k,k+1} = XC(K)$$

$$H_{k,k+I-1} = XA(K)*XGAM(K-1)$$

$$H_{k,k-I+1} = XALP(K)*XC(K-IMAX)$$

$$y_k = XH$$

$$z_k = XG(K)$$

$$x_k^{\ell+1} = ENEW$$

ALTERNATING-DIRECTION-IMPLICIT METHOD

The alternating-direction-implicit, or ADI, method for solving elliptic systems of difference equations is usually attributed to Peaceman and Rachford (Ref. 11). Their procedure is, briefly, to solve Eq. (185) alternately by rows and by columns. The variant that was chosen for the comparison test operates in the following way.

Let $E_{i,j}^{\ell}$ be the ℓ^{th} estimate of $E_{i,j}$ and let ω be a parameter. The procedure is first to solve

$$B_{i,j}^{L}\,(E_{i,j}^{\ell+\frac{1}{2}} - E_{i-1,j}^{\ell}) + B_{i,j}^{R}\,(E_{i,j}^{\ell+\frac{1}{2}} - E_{i+1,j}^{\ell+\frac{1}{2}}) + (1+\omega)\,A_{i,j}\,E_{i,j}^{\ell+\frac{1}{2}}$$
$$= D_{i,j} + \omega\,E_{i,j} - B_{i,j}^{A}\,(E_{i,j}^{\ell} - E_{i,j+1}^{\ell}) - B_{i,j}^{B}\,(E_{i,j}^{\ell} - E_{i,j-1}^{\ell}) \tag{206}$$

for $E_{i,j}^{\ell+\frac{1}{2}}$ and then to solve

$$B_{i,j}^{A}\,(E_{i,j}^{\ell+1} - E_{i,j+1}^{\ell+1}) + B_{i,j}^{B}\,(E_{i,j}^{\ell+1} - E_{i,j-1}^{\ell+1}) + (1+\omega)\,A_{i,j}\,E_{i,j}^{\ell+1}$$
$$= D_{i,j} + \omega\,E_{i,j}^{\ell+\frac{1}{2}} - B_{i,j}^{L}\,(E_{i,j}^{\ell+\frac{1}{2}} - E_{i-1,j}^{\ell+\frac{1}{2}}) - B_{i,j}^{R}\,(E_{i,j}^{\ell+\frac{1}{2}} - E_{i+1,j}^{\ell+\frac{1}{2}}) \tag{207}$$

The parameter ω usually falls in the range $0 \le \omega \le 20$.

The system of equations represented by Eq. (206) may be reduced to j_{max} independent subsystems, one for each value of j. Each of the subsystems has the form

$$a_i x_{i-1} + b_i x_i + c_i x_{i+1} = d_i \tag{208}$$

for $1 \le i \le i_{max}$. To solve Eq. (210), look for a solution of the form

$$x_i = \gamma_i x_{i+1} + \delta_i \tag{209}$$

Using Eq. (209) to eliminate x_{i+1} and x_{i-1} from Eq. (208) yields

$$a_i(\gamma_{i-1} x_i + \delta_{i-1}) + b_i x_i + c_i \frac{x_i - \delta_i}{\gamma_i} = d_i$$

or

$$\left(a_i \gamma_{i-1} + b_i + \frac{c_i}{\gamma_i}\right) x_i = d_i + \frac{c_i \delta_i}{V_i} - a_i \delta_{i-1} \tag{210}$$

Setting both sides of Eq. (210) to zero gives

$$\gamma_i = \frac{-c_i}{a_i \gamma_{i-1} + b_i} \tag{211}$$

$$\delta_i = - \frac{i}{c_i} (d_i - a_i \delta_{i-1})$$

$$= \frac{d_i - a_i \delta_{i-1}}{a_i \gamma_{i-1} + b_i} \tag{212}$$

Sine $a_1 = 0$, the values of γ_o and δ_o are irrelevant. Equations (211) and (212) may be solved recursively for γ_i and δ_i, $i = 1, 2, \ldots, I = i_{max}$. Since $C_I = 0$, $\gamma_I = 0$. Hence, Eq. (209) may be solved for x_i $i = I, I - 1, \ldots, 2, 1$, starting with $x_I = \delta_I$.

The system of equations represented by Eq. (207) is of the same type, and the same procedure may be applied to it.

In the FORTRAN program, γ and δ are stored in arrays XGI and XDI, respectively. E^{ℓ} is in ERAD and $E^{\ell+\frac{1}{2}}$ is in EH.

SUCCESSIVE OVERRELAXATION METHOD

The method of successive overrelaxation (SOR) was probably known to Gauss, but Southwell (Ref. 12) deserves credit for reviving it. In its most primitive form, the method reduces to solving Eq. (185) for $E_{i,j}$ using whatever is the best available estimate for $E_{i,j\pm1}$ and $E_{i\pm1,j}$, then extrapolating the result by some constant amount, and then proceeding to some new (i,j). In the present context, SOR refers to the process of successively solving Eq. (185) for various subsets of the set of all the E's, using the latest estimates of the values of the remaining ones.

To be more specific, let $i_{1,n}$ and $i_{2,n}$ be given for $1 \le n \le N$, where $1 \le i_{1,n} \le i_{2,n} \le i_{max}$. One iteration of the procedure consists of solving N sets of equations, where set n consists of Eq. (185) for $i_{1,n} \le i \le i_{2,n}$ and $1 \le j \le j_{max}$. Of course, if $i_{1,1} = 1$ and $i_{2,1} = i_{max}$, then the first step of the first iteration would produce the exact solution. Usually, the number of columns

$$\Delta i_n = i_{2,n} - i_{1,n} + 1 \tag{213}$$

lies between 1 and 5.

The procedure for carrying out step n of an iteration is as follows. First, the residuals $r_{i,j}$ of Eq. (185) are calculated,

$$r_{i,j} = D_{i,j} - A_{i,j}\, E_{i,j} - B^{L}_{i,j}\,(E_{i,j} - E_{i-1,j}) - \ldots, \tag{214}$$

for $1 \le j \le j_{max}$ and $i_{1,n} \le i \le i_{2,n}$, using the latest values for the E' s. Then corrections $\epsilon_{i,j}$ are calculated using Eq. (185) with $\epsilon_{i,j}$ in place of $E_{i,j}$ and $r_{i,j}$ in place of $D_{i,j}$. Equation (185) then reads

$$B^L_{i,j}(\epsilon_{i,j} - \epsilon_{i-1,j}) + B^R_{i,j}(\epsilon_{i,j} - \epsilon_{i+1,j}) + B^A_{i,j}(\epsilon_{i,j} - \epsilon_{i,j+1})$$
$$+ B^B_{i,j}(\epsilon_{i,j} - \epsilon_{i,j-1}) + A_{i,j}\,\epsilon_{i,j} = r_{i,j} \tag{215}$$

where again, $i_{1,n} \leq i \leq i_{2,n}$. If i is outside that range, then $\epsilon_{i,j} = 0$, so there are not an excessive number of unknowns. The ϵ's having been found, the E's are updated by

$$E^{new}_{i,j} = E^{old}_{i,j} + \omega\,\epsilon_{i,j} \tag{216}$$

where ω is an acceleration parameter. The method used for solving Eq. (215) is Gaussian elimination. Rewrite Eq. (215) in the form

$$\sum b_{k,m}\, e_m = d_k \tag{217}$$

where

$$\left.\begin{aligned}
e_k &= \epsilon_{i,j} \\
d_k &= r_{i,j} \\
b_{k,k} &= A_{i,j} + B^L_{i,j} + B^R_{i,j} + B^A_{i,j} + B^B_{i,j} \\
b_{k,k-1} &= -B^L_{i,j} \quad (i \neq i_{1,n}) \\
b_{k,k+1} &= -B^R_{i,j} \quad (i \neq i_{2,n}) \\
b_{k,k-1} &= -B^B_{i,j} \\
b_{k,k+1} &= -B^A_{i,j}
\end{aligned}\right\} \tag{218}$$

and

$$k = (j - 1)\, t + i + 1 - i_{1,n}$$
$$t = \Delta i_n \tag{219}$$

As in Oliphant's method, $(b_{k,m})$ is expressed as a product of two triangular factors $(w_{k,m})$ and $(v_{k,m})$. Here, the factors are to have the properties

$$w_{k,m} = 0 \quad \text{unless } k - t \le m \le k$$
$$w_{k,k} = 1 \tag{220}$$
$$v_{k,m} = 0 \quad \text{unless } k \le m \le k + t$$

so that

$$b_{k,m} = \sum_{\ell=k-t}^{m} w_{k,\ell}\, v_{\ell,m} \qquad \text{for } m < k \tag{221}$$

$$b_{k,m} = \sum_{\ell=m-t}^{k} w_{k,\ell}\, v_{\ell,m} + v_{k,m} \qquad \text{for } k \le m \tag{222}$$

The solution of Eq. (222) is straightforward. Suppose that $w_{\ell,m}$ and $v_{\ell,m}$ are known for all m when $\ell < k$. Then Eq. (221) can be used to find $w_{k,m}$,

$$w_{k,m} = \frac{b_{k,m} - \sum_{\ell=k-t}^{m-1} w_{k,\ell}\, v_{\ell,m}}{v_{m,m}} \tag{223}$$

for m running from k - t up to k - 1; and then $v_{k,m}$ can be found from Eq. (223):

$$v_{k,m} = b_{k,m} - \sum_{\ell=m-t}^{k-1} w_{k,\ell}\, v_{\ell,m} \tag{224}$$

The w' s and v' s are found once for all iterations. There are

$$\sum_{n=1}^{N} (2t_n + 1)\, t_{n,j_{max}}$$

of them -- too many to hold internally for interesting sizes of t_n. With the w' s and v' s known, the unknown e' s are found by solving $w\,v\,e = b\,e = r$ in two steps: $w\,f = r$, then $v\,e = f$. Thus,

$$f_k = r_k - \sum_{\ell=k-t}^{k-1} w_{k,\ell}\, f_\ell \tag{225}$$

which may be evaluated for increasing k, and then

$$e_k = \frac{f_k - \sum_{\ell=k+1}^{k+t} v_{k,\ell}\, e_\ell}{v_{k,k}} \tag{226}$$

which is evaluated by starting with $k = k_{max} = j_{max}\, t$ and working back down.

The matrices w and v for all steps are packed in the array EF in the FORTRAN program. The matrix elements for step L are assigned to (EF(KF),KF=LB,LC), where LB=LBASE(L) and LC=LBASE(L+1)-1. $w_{k,m}$, for $m < k$, and $v_{k,m}$, for $k \leq m$, are stored in EF(LB+KF), where $KF = (2\ i + 1)\ (k - 1) + m$. The vectors d, e, and f are stored, respectively, in arrays RES, EP, and RES.

THE TESTING PROCEDURE

The three methods described above were exercised on a number of problems. Three of the problems were used for detailed tests reported in the next section.

To describe the tests themselves, it is convenient to use the matrix notation $B\,x = s$ of Eq. (188) to designate the system of equations to be solved, in which case the successive estimates of the solution are given by Eq. (191), where the matrix C depends on one or two parameters as well as the type of test. Let

$$d^{\ell} = x^{\ell+1} - x^{\ell} = C(s - B\,x^{\ell}) \tag{227}$$

be the difference of two successive iterates. Then one can use

$$\delta_{\ell} = \left(\sum_{k} (d_k^{\ell})^2\right)^{\frac{1}{2}} \tag{228}$$

as a measure of the change on iteration .

The ratio

$$\rho_{\ell} = \delta_{\ell}/\delta_{\ell+1} \tag{229}$$

is a measure of the rate of convergence of the iteration. But

$$\begin{aligned} d^{\ell} &= x^{\ell+1} - x^{\ell} \\ &= y - e^{\ell+1} - (y - e^{\ell}) \\ &= E\,e^{\ell-1} - E\,e^{\ell} \\ &= E\,d^{\ell-1} = E^{\ell}\,d^{o} \end{aligned}$$

in the notation used earlier. Consequently, the sequence of difference vectors d^{ℓ} can be expected to converge to an eigenvector of the error matrix E, and ρ_{ℓ} will generally converge to the reciprocal of the modulus of its largest eigenvalue. Asymptotically, at least, $1/\log \rho$ is the number of iterations required to get an e-fold reduction of the error; and if T is the computing time for an iteration, then

*See the section entitled "Common Features of the Method."

$$\phi = \frac{\log \rho}{T} \tag{230}$$

is the rate of convergence in e-folds per unit of time.

TEST RESULTS

The various methods described above were exercised on a number (~10) of test problems. They all performed well on problems containing no zones less than 10 mfp in diameter, and they all converged slowly, if at all, on problems containing many zones of diameter ~0.1 mfp. Three problems of the latter type were used as a basis for a detailed comparison.

Problem 1: $i_{max} = j_{max} = 20$

$r_i - r_{i-1} = 0.1$, for all i; $z_j - z_{j-1} = 0.1$, for all j; $\sigma_{i,j} = 1$, for i and j.

Problem 2 is the same as Problem 1 except that $\sigma_{i,j} = 0.1$ for all i and j.

Problem 3: $i_{max} = j_{max} = 20$.

$$r_i - r_{i-1} = \begin{cases} 0.01 & \text{for } i \le 10 \\ 0.1 & \text{for } 10 < i \end{cases}$$

$$z_j - z_{j-1} = \begin{cases} 0.01 & \text{for } j \le 10 \\ 0.1 & \text{for } 10 < j \end{cases}$$

$$_{i,j} = 0.1, \quad \text{for all } i \text{ and } j$$

The test results are summarized in tables I, II, and III. The parameters that were varied are those described earlier in the sections entitled "Oliphant's Method," "Alternating-Direction-Implicit Method," and "Successive Overrelaxation Method." ADI appears not to perform as well as the other two on easy problems but did quite well on the hardest problem (3). It also seemed to be fairly insensitive to the parameter ω so long as

it fell in the range $5 \leq \omega \leq 10$. Oliphant's method performed very well on easy problems but could not be made to work on Problem 3. (Detailed checking of intermediate results were made to ensure the accuracy of the test of Oliphant's method applied to Problem 3.) The value of ϕ in ADI could be increased by storing rather than regenerating the array of γ's, which does not change after the first iteration, but Oliphant's method would still converge more rapidly on problems like Problem 1. SOR performs best of all on all problems relative to the criterion of largest ϕ. The value of ϕ is based on the time required to compute an iteration but does not account for the preliminary time spent in calculating the approximate inverse, which is very high for high Δi. Consequently, SOR would not perform well on practical problems that converge in a few (~ 10) iterations. Another difficulty with SOR is its large storage requirement, which would make it necessary to use I/O files on large problems.

It is not entirely clear what conclusions can be drawn from the test results. It can be said that Oliphant's method is about as good as any on problems to which TDRAD can practically be applied. ADI is perhaps a better choice because of its greater reliability and insensitivity to the value of ω. SOR, while it works well on hard problems, requires the choice of the sequences $i_{1,n}$ and $i_{2,n}$ and is slow on easy problems because of the setup time it requires.

One fact stands out: none of the methods perform well on problems that involve streaming through thin regions. It may well be that in two or three dimensions, discrete ordinate methods are more efficient than nonequilibrium diffusion when streaming is important.

TABLE I

TEST OF ADI

Problem	ω	ρ	ϕ
1	0	1.026	0.313
	1	1.057	0.663
	3	1.122	1.30
	5	1.191	2.03
	7.5	1.28	2.9
	10	1.21	2.3
2	1	1.019	0.23
	3	1.022	0.24
	5	1.021	0.25
	7.5	1.022	0.26
	10	1.019	0.23
	12	1.017	0.19
3	1	0.962	-0.45
	3	1.026	0.31
	5	1.043	0.50
	7.5	1.072	0.83
	10	1.138	1.54
	12	1.119	1.34

TABLE II

TEST OF OLIPHANT'S METHOD

Problem	λ	ω	ρ	φ
1	0.2	0.9	1.09	1.73
		1.0	1.13	2.45
		1.1	1.22	3.85
		1.2	1.25	4.37
		1.3	0.98	-0.35
	0.3	0.7	1.06	1.14
		0.8	1.08	1.61
		0.9	1.13	2.43
		1.0	1.22	4.02
		1.1	1.32	5.64
		1.2	1.00	0.94
	0.4	0.7	1.07	1.46
		0.8	1.12	2.25
		0.9	1.21	3.56
		1.0	1.38	6.38
		1.1	0.96	-0.09
	0.5	0.6	1.06	1.25
		0.7	1.10	1.95
		0.8	1.18	3.32
		0.9	1.34	5.91
		1.0	0.92	-1.67
		1.1	0.70	-7.27
2	0.2	1.0	1.01	0.16
	0.3	1.0	1.01	0.25
	0.4	0.9	1.01	0.26
		1.0	1.03	0.56
		1.1	0.82	-3.9
	0.5	1.0	0.43	-17
3	-1	0.8	1.009	0.17
		1.0	1.009	0.18
		1.2	1.008	0.18
	-0.8	0.8	1.009	0.17
		1.0	1.008	0.17
		1.2	1.007	0.15
	-0.6	1.0	1.007	0.14
	-0.4	1.0	1.006	0.13
	-0.2	1.0	1.006	0.12
	0	1.0	1.004	0.08

TABLE III

TEST OF SOR

Problem	N	Δi	ω	ρ	φ
1	6	5	1.0	1.48	2.7
			1.1	1.67	3.5
			1.2	2.04	4.9
			1.25	2.52	6.4
2	6	5	1.2	1.043	0.29
			1.3	1.053	0.35
			1.4	1.069	0.45
			1.5	1.092	0.57
	3	10	1.2	1.24	0.94
			1.3	1.30	1.2
			1.4	1.41	1.6
			1.5	1.79	2.7
			1.6	1	0
3	6	5	1.2	1.039	0.26
			1.3	1.049	0.34
			1.5	1.085	0.58
			1.8	1	0
	3	10	1.2	1.27	1.12
			1.3	1.33	1.3
			1.5	1.59	2.2
			1.8	1	0

This page intentionally left blank.

APPENDIX

LONG2, SHORT2, AND NONEQUILIBRIUM DIFFUSION ROUTINES

This page intentionally left blank.

LONG2 ROUTINES

Subroutine DRAW
Subroutine TRAN2

This page intentionally left blank.

Subroutine DRAW

```
C      DRAW
       SUBROUTINE DRAW
C     COMPILED  16NOV66,WBL          29SEPT67,PRW
C
C
C      **************************************************************
C          H E C T I C   C O M M O N
C      **************************************************************
C
       COMMON  AHN,     AMBILN, ANN,     ATOM,    BCATAG, BCBTAG, BCLTAG,
      1BCRTAG, CAPIN,   CAPS,   CDUT,    CMXK,    CMXOM,  CNDE,   CO,
      2CUE,    CSTOP,   CV,     CYCLE,   DBGPRT,  DGREY,  DHNU,   DIANTP,
      3DMIN,   DNN,     DT,     DTBUGR,  DTC,     DTH,    DTNA,   DTO,
      4DTR,    DTR1,    DTR2,   DTUF,    DTVF,    DUMPT7, ECK,    EFRAC,
      5EII,    ERRCRT,  ETH,    EZERO,   FDTC,    FFA,    FFB,    FIFT,
      6FUDGE,  GV,      HCB,    HCP,     HH,      HNU,    HNUP,   HNUS,
      7HVB,    I,       IC,     IGOTO,   IH,      IHNU,   IMAX,   IMPTAG,
      8ISEND,  ISR,     ITAG,   ITRMAX,  IU,      IV,     I1,     I2,
      9J,      JC,      JH,     JMAX,    JMAXA,   JU,     JV,     J5
       COMMON  K,       KMAXA,  KMN1,    KMN2,    KP,     L,      M,
      1MERGE,  MFTAG,   MZ,     N,       NC,      NHNU,   NK,     NPC,
      2NR,     NRM,     NT,     NTAPE,   NY,      N1,     N2,     N3,
      3N4,     ODUC,    PABOVE, PBLO,    PI,      PIDTS,  PRINTL, PRINTS,
      4PROB,   PRR,     PW,     RADE,    RC,      RFT,    RPTAG,  RR,
      5SCDR,   SCR,     SCRE,   SCYCLE,  SGNL,    SIG,    SIGH,   SLUG,
      6SN,     SPROB,   SVMAX,  SVS,     S1,      S2,     S3,     S4,
      7SS,     T,       TAUDTS, THICK,   TMAX,    TRAD,   URR,    UT,
      8UU,     UVMAX,   VABOVE, VAPE,    VBLO,    VEL,    WS,     WSA,
      9XMAX,            BLANK(49)
       COMMON           AIX(1200),       AMX(1200),       ALAMH(1200),
      1ALAMV(1200),     B(1200),         CAP(1200),       ER(1200),
      2FIOUT(1200),     KFIT(1200),      P(1200),         SMLQ(1200),
      3THETA(1200),     U(1200),         V(1200),         XALP(1200),
      4XB(1200),        SOLID(400),      PL(200),         Y(101),
      5DY(100),         GAMC(100),       YAMC(100),       UL(100),
      6W2(100),         X(53),           DX(52),          TAU(52),
      7BETAB(50),       TEMP(12),        HEAD(12)
C
C     ***************************************************************
C
      DIMENSION FLEFT(100), PR(100), SIGC(100)
      EQUIVALENCE (FLEFT, UL), (PL, PR)
      EQUIVALENCE (PL(101), SIGC)
      EQUIVALENCE (S20, BLANK(12)), (ZP126, BLANK(13)), (ZPI36, BLANK(14
     * )), (KMX, BLANK(15)), (WT, BLANK(16)), (ANUMBR, BLANK(19)), (LOCZ
     * , BLANK(22)), (H, BLANK(25)), (NA, BLANK(26))
      EQUIVALENCE (BACC,BLANK(47)),(SACC,BLANK(48)),(TACC,BLANK(49))
C
      REAL MERGE
C
C
      DIMENSION OLDTH(1), PLANCK(1), ROSS(1), XA(1)
      DIMENSION RX(50), RPT(100), KXA(1), NER(1)
      EQUIVALENCE (OLDTH, FIOUT), (PLANCK, P), (ROSS, U), (ER, NER)
      EQUIVALENCE (KXA, XA, V)
      COMMON /BLACK/BETAR(50), BETAA(50)
      COMMON /RADUAT/CLAMDA, ATHETA(50), RTHETA(50), BTHETA(50)
```

```
C     ATHETA,RTHETA,BTHETA CONTAIN BOUNDARY TEMPERATURES (IN EV) FOR THE   DRAW 510
C     ABOVE ,RIGHT,AND BOTTOM BOUNDARIES RESPECTIVELY                      DRAW 520
C                                                                          DRAW 530
      IITOT = 0                                                            DRAW 570
      REWIND 4                                                             DRA8054
      WRITE(4) FIOUT,P,U,V
      REWIND 4
      KMAX=KMAXA-1                                                         DRA8056
      REWIND 3                                                             DRA8057
      DO 10 K=2,KMAX                                                       DRA8058
   10 XALP(K)=0.                                                           DRA8061
      DO 20 I=1,IMAX                                                       DRA8062
      BETAA(I) = 0.
   20 BETAB(I)=0.                                                          DRA8063
      DO 25 J=1,JMAX
   25 BETAR(J) = 0.
      SMU=1.0
      READ (5,960) CLAMDA
      READ (5,960) (ATHETA(I),I=1,IMAX)
      READ (5,960) (RTHETA(J),J=1,JMAX)
      READ (5,960) (BTHETA(I),I=1,IMAX)
      WRITE (6,9003) CLAMDA
      WRITE (6,9004) (ATHETA(I),I=1,IMAX)
      WRITE (6,9005) (RTHETA(J),J=1,JMAX)
      WRITE (6,9006) (BTHETA(I),I=1,IMAX)
      READ 960, ANGN
      NANG=ANGN
      WRITE (6,1043) NANG
   30 IF (X(0).LT.1.E-10) GO TO 40                                         DRA8073
      S1=5.0025                                                            DRA8074
      GO TO 955
   40 NHGP=0                                                               DRA8076
      IGWD=2                                                               DRA8077
C
C     G R I D   L O O P
C
      DO 910 IANG=1,NANG
      READ  960, EMU,DMU,DZ1,DR1
      WRITE (6,1060) EMU,DMU,DZ1,DR1
      IF (EMU .LT. .99995) GO TO 45
      S1 = 5.0021
      GO TO 955
C     THETA=0 PROHIBITED AS AN INPUT CASE
   45 ETA = SQRT(1.0-EMU*EMU)
      SMU=SMU-DMU
      IF (SMU .GE. 0.) GO TO 47
      S1 = 5.0022
      GO TO 955
   47 COTAN = EMU/ETA
      DXF = 1.0/ETA
      DZF = 1.0/EMU
      IRX=X(IMAX)/DR1+0.8
C     0.8 IS AN ALEXANDERISM
      IF (IRX .LE. 100) GO TO 48
      S1 = 5.0023
```

```
      GO TO 955
 48   RPT(1) = 0.0
      DO 50 IIR=2,IRX
   50 RPT(IIR)=RPT(IIR-1)+DR1
      ZXTRA=X(IMAX)*COTAN
C   FIRST.SET UP LIKE NEITHER TOP NOR BOTTOM IS REFLECTING
      JZX=(Y(JMAX)-Y(0)+ZXTRA)/DZ1+0.5
      ZSTRT=Y(JMAX)+(0.5-FLOAT(JZX))*DZ1
      ZEND=Y(JMAX)+ZXTRA
      IF(BCBTAG .LT. 0.0) ZSTRT=Y(0)+0.5*DZ1
      IF(BCATAG .LT. 0.0) ZEND=Y(JMAX)
      IF (BCBTAG .GT. -1.E-20  .OR.   BCATAG .GT. -1.E-20) GO TO 95
      S1 = 5.0024
      GO TO 955
 95   WT = UMU * DZ1 * DR1 * ETA
C
C   C O M B   L O O P
C
      DO 900 IRD=1,IRX
      IF(IRD .GT. 2)  GO TO 100
      IF (IRD .EQ. 1)  WT = 0.5*WT
      IF (IRD .EQ. 2) WT = 2.0*WT
  100 CONTINUE
      PARIM=RPT(IRD)                                                    DRA8159
      IQ=0                                                              DRA8160
      DO 290 I=1,IMAX                                                   DRA8161
      QW=X(I)**2-PARIM**2                                               DRA8162
      IF (QW.LT.1.E-10) GO TO 290                                       DRA8163
      IQ=IQ+1                                                           DRA8164
      RX(IQ)=SQRT(QW)                                                   DRA8165
  290 CONTINUE                                                          DRA8166
      IF (IQ.EQ.0) GO TO 900                                            DRA8167
      IRSTRT=IMAX-IQ+1                                                  DRA8168
C
C   R A Y   L O O P
C
      ZCNTR=ZSTRT-DZ1
      DO 890 IZRAY=1,1000
      ZCNTR=ZCNTR+DZ1
      IF(ZCNTR .GT. ZEND) GO TO 900
      ITOP=0                                                            DRA8171
      KXA(1)=0                                                          DRA8172
      KXA(201)=0                                                        DRA8173
C   B E G I N   U P W A R D   T R A C E
      IUP=1                                                             DRA8176
      IUPUP=1                                                           DRA8177
      IF (ZCNTR.LT.Y(0)+1.E-5.AND.BCBTAG.LT.0.) GO TO 890               DRA8178
      IF (ZCNTR.LT.Y(0)) GO TO 300                                      DRA8179
      GO TO 340                                                         DRA8180
C   CENTER OF RAY IS BELOW THE SYSTEM - - INITIALIZE FOR UPWARD TRACE
  300 SDELZ=Y(0)-ZCNTR                                                  DRA8183
      IF (COTAN.LT.1.E-10) GO TO 730                                    DRA8184
      SDELX=SDELZ/COTAN                                                 DRA8185
      IF (SDELX.GT.RX(IQ)) GO TO 730                                    DRA8186
      IQQ=IQ-1                                                          DRA8187
```

```
  310 IF (IQQ.EQ.0.OR.SDELX.GT.RX(IQQ)) GO TO 320                          DHA8188
      IQQ=IQQ-1                                                             DHA8189
      IF (IQQ.GT.0) GO TO 310                                               DHA8190
      ITR=IRSTRT                                                            DHA8191
      GO TO 330                                                             DHA8192
  320 ITR=IQQ+IRSTRT                                                        DHA8193
  330 JTR=1                                                                 DHA8194
      ZTR=Y(1)                                                              DHA8195
      BETAB(ITR) = BETAB(ITR) + WT
      XTR=SDELX                                                             DHA8197
      GO TO 380                                                             DHA8198
  340 IF (ZCNTR.LT.Y(JMAX)) GO TO 350                                       DHA8199
      ITOP=1                                                                DHA8200
      GO TO 640                                                             DHA8201
C   CENTER OF RAY IS IN SYSTEM - - INITIALIZE FOR UPWARD TRACE
  350 DO 370 J=1,JMAX                                                       DHA8203
      IF (ZCNTR.LT.Y(J)) GO TO 360                                          DHA8204
      GO TO 370                                                             DHA8205
  360 ZTR=ZCNTR                                                             DHA8206
      XTR=0.                                                                DHA8207
      ITR=IRSTRT                                                            DHA8208
      JTR=J                                                                 DHA8209
      GO TO 380                                                             DHA8210
  370 CONTINUE                                                              DHA8211
      SI=5.0075                                                             DHA8212
      GO TO 955
C   G E N E R A L   C A S E   O F   I N C R E A S I N G   Z
  380 DELZ=Y(JTR)-ZTR                                                       DHA8215
      ICOR = 0
      KTR=ITR+1+IMAX*(JTR-1)                                                DHA8216
      IXX=ITR-IRSTRT+1                                                      DHA8217
      IF (IXX.GT.0) GO TO 390                                               DHA8218
      SI=5.008                                                              DHA8219
      GO TO 955
  390 DELX=RX(IXX)-XTR                                                      DHA8221
  400 IF (DELZ-DELX*COTAN) 410,430,420                                      DHA8227
C   RAY HITS TOP OF CELL
  410 DELS=DELZ*DZF                                                         DHA8229
      ZTR=Y(JTR)                                                            DHA8230
      JTR=JTR+1                                                             DHA8232
      XTR=XTR+DELZ/COTAN                                                    DHA8233
      GO TO 440                                                             DHA8234
C   RAY HITS SIDE OF CELL
  420 DELS=DELX*DXF                                                         DHA8236
      XTR=RX(IXX)                                                           DHA8237
      ZTR=ZTR+DELX*COTAN                                                    DHA8238
      ITR=ITR+1                                                             DHA8240
      GO TO 440                                                             DHA8241
C   RAY HITS CORNER OF CELL
  430 DELS=DELX*DXF                                                         DHA8243
      XTR=RX(IXX)                                                           DHA8244
      ZTR=Y(JTR)                                                            DHA8245
      JTR=JTR+1                                                             DHA8248
      ITR=ITR+1                                                             DHA8249
      ICOR = 1
```

```
C     STORE THE PATH LENGTH AND CELL INDEX                                  DRA8251
  440 KXA(IUP)=KTR                                                          DRA8252
      XB(IUP)=DELS                                                          DRA8253
      IUP=IUP+1
C     CHECK TO SEE IF, AND WHERE, RAY HITS SYSTEM BOUNDARY                  DRA8255
      IF (ITR.LE.IMAX) GO TO 460
C     RAY HITS SIDE BOUNDARY
      IF (ICOR .EQ. 0) GO TO 446
      BETAR(JTR-1) = BETAR(JTR-1) + WT
      GO TO 449
  446 BETAR(JTR) = BETAR(JTR) + WT
  449 IF (IUP.LE.200)  GO TO 450                                            DRA8257
      NXIT=2                                                                DRA8258
      GO TO 640                                                             DRA8259
  450 NTER=2                                                                DRA8260
      GO TO 640                                                             DRA8261
  460 IF (JTR.LE.JMAX) GO TO 380
C     RAY HITS TOP BOUNDARY                                                 DRA8262
      IF (BCATAG.LT.0.) GO TO 480
      BETAA(ITR) = BETAA(ITR) + WT                                          DRA8263
      IF (IUP.LE.200) GO TO 470                                             DRA8264
      NXIT=3                                                                DRA8265
      GO TO 640                                                             DRA8266
  470 NTER=3                                                                DRA8267
      GO TO 640
C     RAY HITS TRANSMITTIVE TOP BOUNDARY                                    DRA8268
  480 IF (IUP.LE.200) GO TO 490                                             DRA8269
      S1=5.0105
      GO TO 955
C     RAY HITS REFLECTIVE TOP BOUNDARY -- INITIALIZE FOR DOWNWARD TRACE     DRA8272
C     GO DOWN IN Z.                                                         DRA8273
  490 ZTR=Y(JMAX)                                                           DRA8274
      JTR=JMAX
C     G E N E R A L   C A S E   O F   D E C R E A S I N G   Z
      ICOR = 0                                                              DRA8276
  500 IXX=ITR-IRSTRT+1                                                      DRA8277
      MZCRS=1                                                               DRA8278
      IF (IXX.GT.0) GO TO 510                                               DRA8279
      S1=5.0112
      GO TO 955                                                             DRA8281
  510 DELX=RX(IXX)-XTR                                                      DRA8282
      DELZ=ZTR-Y(JTR-1)                                                     DRA8283
      KTR=ITR+1+IMAX*(JTR-1)                                                DRA8291
  520 IF (DELZ-DELX*COTAN) 530,550,540
C     RAY HITS BOTTOM OF CELL                                               DRA8293
  530 DELS=DELZ*DZF                                                         DRA8294
      ZTR=Y(JTR-1)                                                          DRA8295
      JTR=JTR-1                                                             DRA8296
      XTR=XTR+DELZ/COTAN                                                    DRA8297
      MZCRS=2                                                               DRA8298
      GO TO 560
C     RAY HITS SIDE OF CELL                                                 DRA8300
  540 DELS=DELX*DXF                                                         DRA8301
      XTR=RX(IXX)                                                           DRA8302
      ZTR=ZTR-DELX*COTAN
```

```
      ITR=ITR+1                                                         DHA8304
      GO TO 560                                                         DHA8305
C   RAY HITS CORNER OF CELL
  550 DELS=DELX*DXF                                                     DHA8307
      XTR=XA(IXX)                                                       DHA8308
      ZTR=Y(JTR-1)                                                      DHA8309
      ALAMV(KTR)=ALAMV(KTR)+1.                                          DHA8310
      JTR=JTR-1                                                         DHA8311
      ITR=ITR+1                                                         DHA8312
      ICOR = 1
      MZCRS=2                                                           DHA8313
C   STORE THE PATH LENGTH AND CELL INDEX
  560 KXA(IUP)=KTR                                                      DRA8315
      XB(IUP)=DELS                                                      DHA8316
      IUP=IUP+1                                                         DHA8317
C   CHECK TO SEE IF, AND WHERE, RAY HITS SYSTEM BOUNDARY
      IF (ITR.LE.IMAX) GO TO 580                                        DHA8319
C   RAY HITS SIDE BOUNDARY
      IF (ICOR .EQ. 0) GO TO 566
      BETAR(JTR+1) = BETAR(JTR+1) + WT
      GO TO 569
 566  BETAR(JTR) = BETAR(JTR) + WT
 569  IF (IUP .LE. 200) GO TO 570
      NXIT=2                                                            DHA8321
      GO TO 640                                                         DHA8322
  570 NTER=2                                                            DHA8323
      GO TO 640                                                         DHA8324
  580 IF (JTR.LT.1) GO TO 600                                           DHA8325
      GO TO (500,590), MZCRS                                            DHA8326
  590 M=KTR-IMAX                                                        DHA8327
      GO TO 500                                                         DHA8329
C   RAY HITS BOTTOM BOUNDARY
 600  IF (BCBTAG .LT. 0.) GO TO 620
C   RAY HITS TRANSMITTIVE BOTTOM BOUNDARY
      BETAB(ITR) = BETAB(ITR) + WT
      IF (IUP.LE.200) GO TO 610                                         DHA8332
      NXIT=1                                                            DHA8333
      GO TO 640                                                         DHA8334
  610 NTER=1                                                            DHA8335
      GO TO 640                                                         DHA8336
  620 IF (IUP.GT.200) GO TO 630                                         DRA8337
      S1=5.0135                                                         DHA8338
      GO TO 955
C   RAY HITS REFLECTIVE BOTTOM BOUNDARY - - INITIALIZE FOR UPWARD TRACE
  630 ZTR=Y(0)                                                          DHA8341
      JTR=1                                                             DHA8342
      GO TO 380                                                         DHA8343
C   B E G I N   D O W N W A R D   T R A C E
  640 IF (IUP.GT.200) GO TO 730                                         DHA8345
      IUPUP=IUP-1                                                       DHA8346
      IUP=201                                                           DHA8347
      IF (ZCNTR.GT.Y(JMAX)-1.E-5.AND.BCATAG.LT.0.) GO TO 890            DHA8348
      IF (ITOP.GT.0) GO TO 650                                          DHA8349
      GO TO 690                                                         DHA8350
C   CENTER OF RAY IS AT TOP OR OUT OF SYSTEM AND NO UPWARD TRACE
```

```
C    WAS PERFORMED                                                      DRA8352
  650 SDELZ=ZCNTR-Y(JMAX)                                               DRA8353
      IF (COTAN.LT.1.E-10) GO TO 730                                    DRA8354
      SDELX=SDELZ/COTAN                                                 DRA8355
      IF (SDELX.GT.RX(IQ)) GO TO 730                                    DRA8356
      IQQ=IQ-1                                                          DRA8357
  660 IF (SDELX.GT.RX(IQQ)) GO TO 670                                   DRA8358
      IQQ=IQQ-1                                                         DRA8359
      IF (IQQ.GT.0) GO TO 660                                           DRA8360
      ITR=IRSTRT                                                        DRA8361
      GO TO 680                                                         DRA8362
  670 ITR=IQQ+IRSTRT                                                    DRA8363
  680 JTR=JMAX                                                          DRA8364
      ZTR=Y(JMAX)                                                       DRA8365
      XTR=SDELX                                                         DRA8366
      KTR=ITR+1+IMAX*(JTR-1)
      BETAA(ITR) = BETAA(ITR) + WT                                      DRA8368
      GO TO 500                                                         DRA8369
  690 IF (ZCNTR.GT.Y(0)) GO TO 700                                      DRA8370
      GO TO 730
C    CENTER OF RAY IS INSIDE SYSTEM AND AN UPWARD TRACE WAS PERFORMED -
C    - INITIALIZE FOR DOWNWARD TRACE                                    DRA8372
  700 DO 720 J=1,JMAX                                                   DRA8373
      JB=JMAX+1-J                                                       DRA8374
      IF (ZCNTR.GT.Y(JB-1)) GO TO 710                                   DRA8375
      GO TO 720                                                         DRA8376
  710 ZTR=ZCNTR                                                         DRA8377
      XTR=0.                                                            DRA8378
      ITR=IRSTRT                                                        DRA8379
      JTR=JB                                                            DRA8380
      GO TO 500                                                         DRA8381
  720 CONTINUE                                                          DRA8382
      S1=5.0160
      GO TO 955
C
C    T R A C E   O F   A   R A Y   I S   C O M P L E T E
C    (EXCEPT THETA=0 TRACE)
C
C
C    MERGE STARTING SEGMENTS IF IN SAME CELL                            DRA8386
  730 IF (KXA(1).LE.0.AND.KXA(201).LE.0) GO TO 890                      DRA8387
      IF (KXA(1).EQ.KXA(201)) GO TO 740                                 DRA8388
      IDNMN=201                                                         DRA8389
      GO TO 750                                                         DRA8390
  740 IDNMN=202                                                         DRA8391
      XB(1)=XB(1)+XB(201)
C    CONTRIBUTE TO SUMMED PATH LENGTHS FOR EACH CELL                    DRA8393
  750 L1=1                                                              DRA8394
      L2=IUPUP                                                          DRA8395
      L3=1                                                              DRA8396
      IUP=IUP-1                                                         DRA8397
  760 DO 770 II=L1,L2                                                   DRA8398
      K=KXA(II)
  770 XALP(K) = XALP(K) + XB(II)*WT                                     DRA8400
      GO TO (780,790), L3
```

```
  780 IF (IDNMN.GT.IUP) GO TO 790                                        DRA8401
      L1=IDNMN                                                           DRA8402
      L2=IUP                                                             DRA8403
      L3=2                                                               DRA8404
      GO TO 760                                                          DRA8405
C   TURN THE UPWARD DATA AROUND
  790 IUPM=IUPUP/2                                                       DRA8407
      IF (IUPM.LE.0) GO TO 810                                           DRA8408
      DO 800 I=1,IUPM                                                    DRA8409
      IB=IUPUP+1-I                                                       DRA8410
      KXI=KXA(I)                                                         DRA8411
      XX2=XB(I)                                                          DRA8412
      KXA(I)=KXA(IB)                                                     DRA8413
      XB(I)=XB(IB)                                                       DRA8414
      KXA(IB)=KXI                                                        DRA8415
  800 XB(IB)=XX2                                                         DRA8416
  810 IF (IUPUP.LT.1.AND.IUP.LT.IDNMN) GO TO 890                         DRA8417
      IF (IUP.LT.IDNMN) GO TO 830                                        DRA8418
C   MOVE DOWNWARD DATA SNUG NEXT TO UPWARD DATA
      DO 820 I=IDNMN,IUP                                                 DRA8420
      INW=IUPUP+1-IDNMN+I                                                DRA8421
      KXA(INW)=KXA(I)                                                    DRA8422
  820 XB(INW)=XB(I)                                                      DRA8423
      II2=IUPUP+IUP-IDNMN+1                                              DRA8424
      ITOT=II2                                                           DRA8425
      IF (IUPUP.LT.1) NTER=3                                             DRA8426
      GO TO 840                                                          DRA8427
  830 II2=IUPUP                                                          DRA8428
      IF (ZCNTR .LT. Y(0))  NXIT =1
      ITOT=IUPUP                                                         DRA8430
  840  NPOINT = 1
      GO TO 2359
C   DEBUG EDIT
  880  IF (ABS(DBGPRT) .LT. 1.E-20) GO TO 890
      WRITE(6,970)EMU,IRD,IZRAY,NTER,NXIT,WT
      WRITE (6,980) (KXA(I),XB(I),I=1,II2)                               DRA8453
  890 CONTINUE                                                           DRA8454
C   E N D   O F   R A Y   L O O P
  900 CONTINUE                                                           DRA8455
C   E N D   O F   C O M B   L O O P
      WRITE (6,9001)  IITOT
  910 CONTINUE                                                           DRA8456
C   E N D   O F   G R I D   L O O P
C
C   T H E T A  =  0   T R A C E
C
 2349 WTP = SMU * 0.5
      WRITE (6,9007) SMU
      DO 2351 J=1,JMAX
      XB(J) = DY(J)
      KXA(J) = 1 + IMAX * (J-1)
C   I =0  IN ABOVE STATEMENT
      DO 2350 I=1,IMAX
      K = I + 1 + IMAX * (J-1)
      WT = WTP * TAU(I)
```

```
C     NOTE -- FINAL SUM OF LENGTHS FOR REFLECTING AND NONREFLECTING CASE
C     SHOULD BE EQUAL IN ACCORDANCE WITH NO. OF PASSES IN TRAN2
 2350 XALP(K) = XALP(K) + XB(J)*WT
 2351 CONTINUE
      IF (BCATAG.LT.0.) GO TO 2352
      IF (BCBTAG.LT.0.) GO TO 2354
C     NO REFLECTION
      ITOT = JMAX
      NTER = 1
      NXIT = 3
      DO 2330 I=1,IMAX
      WT = WTP * TAU(I)
      BETAA(I) = BETAA(I) + WT
 2330 BETAB(I) = BETAB(I) + WT
      GO TO 2357
C     TOP REFLECTING
 2352 IEND = JMAX - 1
      XB(JMAX) = XB(JMAX) + XB(JMAX)
      DO 2353 J=1,IEND
      XB(JMAX+J) = XB(JMAX-J)
 2353 KXA(JMAX+J) = KXA(JMAX-J)
      ITOT = JMAX + IEND
      NTER = 1
      NXIT = -1
      DO 2331 I=1,IMAX
      WT = WTP * TAU(I)
 2331 BETAB(I) = BETAB(I) + WT * 2.
      GO TO 2357
C     BOTTOM REFLECTING
 2354 IEND = JMAX - 1
      DO 2355 J=1,IEND
      XB(JMAX+J) = XB(J+1)
 2355 KXA(JMAX+J) = KXA(J+1)
      XB(JMAX) = XB(1) + XB(1)
      KXA(JMAX) = 1
C     I=0  IN ABOVE STATEMENT
      DO 2356 J=1,IEND
      XB(JMAX-J) = XB(JMAX+J)
 2356 KXA(JMAX-J) = KXA(JMAX+J)
      ITOT = JMAX + IEND
      NTER = 3
      NXIT = -1
      DO 2332 I=1,IMAX
      WT = WTP * TAU(I)
 2332 BETAA(I) = BETAA(I) + WT * 2.
 2357 II2 = ITOT
C     COMPUTE KXA(I) AND PACK RAY DATA FOR ITH COMB (WHICH CONTAINS ONLY
C     ONE RAY) IN THE THETA=0 CASE
      IM = 0
 2370 IM = IM + 1
      DO 2358 J=1,ITOT
 2358 KXA(J) = KXA(J) + 1
      WT = WTP * TAU(IM)
      NPOINT = 2
      GO TO 2359
```

```
C  DEBUG EDIT
 911  IF (ABS(DBGPRT).LT.1.E-20) GO TO 912
      WRITE (6,9008) IM,NTER,NXIT,WT
      WRITE (6,980) (KXA(I),XB(I),I=1,II2)
 912  IF (IM.LT.IMAX) GO TO 2370
      WRITE (6,9001)  IITOT
C   WRITE END RECORD ON DATA TAPE
      NER(IGWD+1)=1                                                     DRA8458
      NER(IGWD+2)=1                                                     DRA8459
      NER(IGWD+3)=1                                                     DRA8460
      ER(IGWD+4) = -10.
      NER(IGWD+5) = -10
      ER(IGWD+6) = -10.
      NER(2)=NRGP+1                                                     DRA8463
      IGWD = IGWD + 6
      NER(1)=IGWD                                                       DRA8465
      WRITE (3)  (ER(I),I=1,IGWD)
      END FILE 3                                                        DRA8467
      REWIND 3                                                          DRA8468
C   CALCULATE EFFECTIVE AREAS (XALP)
      IWARN=1                                                           DRA8470
      DO 950 K=2,KMAX                                                   DRA8471
      I=MOD(K-2,IMAX)+1                                                 DRA8472
      J=(K-2)/IMAX+1                                                    DRA8473
      IF (XALP(K).LT.1.E-10) GO TO 920                                  DRA8474
      XALP(K) = TAU(I) * DY(J) * 2. * PI/XALP(K)
C   = EFFECTIVE AREA OF CELL K
C   = 4*PI*VOLUME OF CELL / SUM(LENGTH  * WEIGHT OF RAY)
C   NOTE -- XALP WAS HALF THE SUM OF LENGTHS SINCE EACH RAY TRACE IN
C   DRAW ACCOUNTS FOR TWO RAYS IN SYSTEM
      GO TO 950                                                         DRA8476
 920  XALP(K)=1.E20                                                     DRA8477
      GO TO (930,940), IWARN
 930  IWARN=2                                                           DRA8479
      WRITE (6,1050)                                                    DRA8480
 940  WRITE (6,980) K                                                   DRA8481
 950  CONTINUE                                                          DRA8482
C   FINIALIZE VARIABLES AND PRINT DEBUG IF.CALLED FOR
      READ(4) FIOUT,P,U,V
      REWIND 4
      IF (ABS(DBGPRT) .LT. 1.E-20) GO TO 953
      WRITE (6,990)                                                     DRA8487
      WRITE (6,1000) (XALP(K),K=2,KMAX)                                 DRA8488
 953  CONTINUE
      RETURN                                                            DRA8495
 886  S1 = 5.1111
 955  ISEND =2
      READ(4) FIOUT,P,U,V
      REWIND 4
      CALL EDIT
C
C  P A C K   R A Y   D A T A
C
 2359 NWD = II2 + II2 + 4
      IITOT = IITOT + ITOT
```

```
      IF (IGWD + NWD .GT. 1023)  GO TO 870
  850 NER(IGWD+1)=ITOT                                                  DRA8434
      NER(IGWD+2)=NTER                                                  DRA8435
      NER(IGWD+3)=NXIT                                                  DRA8436
      ER(IGWD+4) = WT
      DO 860 I=1,II2                                                    DRA8437
      ITWD = IGWD + I + I + 3
      NER(ITWD)=KXA(I)                                                  DRA8439
  860 ER(ITWD+1)=XB(I)                                                  DRA8440
      IGWD=IGWD+NWD                                                     DRA8441
      NRGP=NRGP+1                                                       DRA8442
      GO TO 885
  870 NER(1)=IGWD                                                       DRA8445
      NER(2)=NRGP                                                       DRA8446
      WRITE (3)  (ER(I),I=1,IGWD)
      IGWD=2                                                            DRA8448
      NRGP=0                                                            DRA8449
      GO TO 850                                                         DRA8450
 885  GO TO (880,911), NPOINT
C
C
C                                                                       DRA8496
  960 FORMAT (6E12.5)
  970 FORMAT (/6H MU = ,F10.4,6X,8H R INDEX,I5,6X,8H Z INDEX,I5,6X,9H BC
     1 ENTER,I5,6X,8H BC EXIT,I5,4X,3H WT,1PE12.5)
  980 FORMAT (6(I5,1PE15.8))                                            DRA8500
  990 FORMAT (6H1AREAS)                                                 DRA8501
 1000 FORMAT (1P8E15.8)                                                 DRA8502
 1010 FORMAT (50H1CENSUS OF RAYS CROSSING THE OUTER SYSTEM BOUNDARY)
 1020 FORMAT (48H1CENSUS OF RAYS CROSSING THE TOP SYSTEM BOUNDARY)
 1030 FORMAT (51H1CENSUS OF RAYS CROSSING THE BOTTOM SYSTEM BOUNDARY)
 1040 FORMAT (1H0//15HGRID INPUT FOR ,I2,24H GRIDS AND SEGMENT COUNT/)
 1050 FORMAT (42H WARNING.  NO RAYS GO THROUGH THESE ZONES )            DRA8507
 1060 FORMAT (1H ,5X,4HMU= ,1PE12.5,13H     DELTMU= ,E12.5,10H      DZ1=
     1,E12.5,10H      DR1= ,E12.5)
 1070 FORMAT (33H        TOTAL CROSSINGS FOR OUTER =,F10.2)
 1080 FORMAT (34H        TOTAL CROSSINGS FOR BOTTOM =,F10.2)
 1090 FORMAT (32H1TOTAL NUMBER OF RAYS EMPLOYED =,F10.2)
 1100 FORMAT (31H        TOTAL CROSSINGS FOR TOP =,F10.2)
 9001 FORMAT (1H ,5X,47HAFTER COMPLETION OF ABOVE GRID, TOTAL SEGMENTS=,
     ,I10//)
 9003 FORMAT (1H0,10X,8HCLAMDA= ,F12.2)
 9004 FORMAT (1H0,10X,9HATHETA(I)/(1P7E15.5))
 9005 FORMAT (1H0,10X,9HRTHETA(I)/(1P7E15.5))
 9006 FORMAT (1H0,10X,9HBTHETA(I)/(1P7E15.5))
 9007 FORMAT (1H ,5X,8HMU=    1.,13X,8HDELTMU= ,E12.5)
 9008 FORMAT (/12H MU =     1.,7X,7HR INDEX,I5,7X,12HZ INDEX   -1,6X,9H
     1BC ENTER,I5,6X,8H BC EXIT,I5,4X,3H WT,1PE12.5)
      END                                                               DRA8508
```

Subroutine TRAN2

```
C     TRAN2                                                              TRN1000
      SUBROUTINE TRAN2                                                   TRN1001
C    COMPILED  16NOV66,WBL              29SEPT67,PRW
C                                                                        TRN1003
C        H E C T I C   C O M M O N                                       TRN1004
C     ***************************************************************TRN1005
C                                                                        TRN1006
      COMMON  AIIN,   AMBIEN, ANN,    ATOM,   BCATAG, BCBTAG, BCLTAG,
     1BCRTAG, CAPIN,  CAPS,   CDUT,   CMXK,   CMXOM,  CNDE,   CO,
     2COE,    CSTOP,  CV,     CYCLE,  DBGPRT, DGREY,  DHNU,   DIANTP,
     3DMIN,   DNN,    DT,     DTBUGR, DTC,    DTH,    DTNA,   DTQ,
     4DTR,    DTR1,   DTR2,   DTUF,   DTVF,   DUMPT7, ECK,    EFRAC,
     5EI1,    ERRCRT, ETH,    EZERO,  FDTC,   FFA,    FFB,    FIFT,
     6FUDGE,  GV,     HCB,    HCP,    HH,     HNU,    HNUP,   HNUS,
     7HVB,    I,      IC,     IGOTO,  IH,     IHNU,   IMAX,   IMPTAG,
     8ISEND,  ISR,    ITAG,   ITRMAX, IU,     IV,     I1,     I2,
     9J,      JC,     JH,     JMAX,   JMAXA,  JU,     JV,     J5
      COMMON  K,      KMAXA,  KMN1,   KMN2,   KP,     L,      M,
     1MERGE,  MFTAG,  MZ,     N,      NC,     NHNU,   NK,     NPC,
     2NR,     NRM,    NT,     NTAPE,  NY,     N1,     N2,     N3,
     3N4,     ODDC,   PABOVE, PBLO,   PI,     PIDTS,  PRINTL, PRINTS,
     4PROB,   PRR,    PW,     RADE,   RC,     RFT,    RPTAG,  RR,
     5SCDR,   SCR,    SCRE,   SCYCLE, SGNL,   SIG,    SIGH,   SLUG,
     6SN,     SPROB,  SVMAX,  SVS,    S1,     S2,     S3,     S4,
     7SS,     T,      TAUUTS, THICK,  TMAX,   TRAD,   URR,    UT,
     8UU,     UVMAX,  VABOVE, VAPE,   VBLO,   VEL,    WS,     WSA,
     9XMAX,           BLANK(49)
      COMMON          AIX(1200),      AMX(1200),      ALAMH(1200),
     1ALAMV(1200),    B(1200),        CAP(1200),      ER(1200),          HECCOM22
     2FIOUT(1200),    KFIT(1200),     P(1200),        SMLQ(1200),        HECCOM23
     3THETA(1200),    U(1200),        V(1200),        XALP(1200),        HECCOM24
     4XB(1200),       SOLID(400),     PL(200),        Y(101),            HECCOM25
     5DY(100),        GAMC(100),      YAMC(100),      UL(100),           HECCOM26
     6W2(100),        X(53),          DX(52),         TAU(52),           HECCOM27
     7BETAB(50),      TEMP(12),       HEAD(12)                           HECCOM28
C                                                                        TRN1034
C     ***************************************************************TRN1035
C                                                                        TRN1036
      DIMENSION       FLEFT(100),     PR(100),        SIGC(100)          TRN1037
      EQUIVALENCE   (FLEFT,UL) , (PL,PR)
      EQUIVALENCE    (PL(101),SIGC)
      EQUIVALENCE (S20,BLANK(12)), (ZP126,BLANK(13)), (ZP136,BLANK(14)),
     1(KMX,BLANK(15)),(WT,BLANK(16)),(ANUMBR,BLANK(19)),(LOCZ,BLANK(22))
     2,(H,BLANK(25)),(NA,BLANK(26))
      EQUIVALENCE   (BACC,BLANK(47)),(SACC,BLANK(48)),(TACC,BLANK(49))
C                                                                        TRN1041
      REAL  MERGE                                                        TRN1042
C                                                                        TRN1043
C                                                                        TRN1044
      DIMENSION  PUR(50), PUZ(100), RUR(50), RUZ(100)                    TRN1045
      DIMENSION OLDTH(1),PLANCK(1),ROSS(1),XA(1)
      DIMENSION KXA(1)                                                   TRN1047
      DIMENSION  KXB(1)                                                  TRN1048
      EQUIVALENCE (OLDTH,FIOUT),(PLANCK,P),(ROSS,U),
     1 (KXA,XA,V),(KXB,XB)                                               TRN1050
      COMMON  /BLACK/ BETAR(50),BETAA(50)
```

```
      DIMENSION PA(50),PR1(50),PB(50)
      COMMON /RADDAT/ CLAMDA,ATHETA(50),RTHETA(50),BTHETA(50)
C    ATHETA,RTHETA,BTHETA CONTAIN BOUNDARY TEMPERATURES (IN EV) FOR THE
C    ABOVE,RIGHT,AND BOTTOM BOUNDARIES RESPECTIVELY
C     ******************************************************************TRN1051
C     XALP, ALAMV, ALAMH, AND BETAB INVIOLATE -- FORMED IN DRAW         TRN1052
C     NY(IN LINDLEY COMMON) IS TEMPERATURE ITERATION INDEX USED IN KAPPATRN1053
      FOO=0.0                                                           TRN1054
      ETOP=0.0                                                          TRN1055
      ESIDE=0.0                                                         TRN1056
      EBTM=0.0                                                          TRN1057
      KMAX=KMAXA-1                                                      TRN1058
      IF (MFTAG.EQ.0) DHNU=1.                                           TRN1059
      NVEZ=1                                                            TRN1060
      NY=NVEZ                                                           TRN1061
      IF (ITAG.EQ.0) NVEZ=2                                             TRN1062
      VEZ=NVEZ                                                          TRN1063
      CALL DVCHK (KDMY)                                                 TRN1064
      THTAMX=.025                                                       TRN1065
C     CALCULATE GEOMETRY FACTORS AND FIND HIGHEST TEMPERATURE           TRN1066
      DO 10 I=1,IMAX                                                    TRN1067
      PUR(I)=PI/TAU(I)                                                  TRN1068
   10 RUR(I+1)=4.0E10*X(I)/(X(I+1)-X(I-1))                              TRN1069
      DO 20 J=1,JMAX                                                    TRN1070
      PUZ(J)=1./DY(J)                                                   TRN1071
   20 RUZ(J+1)=2.0E10/(Y(J+1)-Y(J-1))                                   TRN1072
      DO 30 K=2,KMAX                                                    TRN1073
      IF (JMR(KFIT(K),2).NE.1) GO TO 30                                 TRN1074
      IF (THETA(K).LT.THTAMX) GO TO 30                                  TRN1075
      THTAMX=THETA(K)                                                   TRN1076
   30 CONTINUE                                                          TRN1077
      THTAMX=AMAX1(THTAMX,BCLTAG,BCRTAG,BCBTAG,BCATAG)                  TRN1078
C     REENTRY POINT FOR SECOND TEMPERATURE ITERATION                    TRN1079
   40 DO 50 K=2,KMAX                                                    TRN1080
      XA(K)=0.                                                          TRN1081
      XB(K)=0.                                                          TRN1082
   50 ER(K)=0.0                                                         TRN1083
C                                                                       TRN1084
C                                                                       TRN1085
C     ******************************************************************TRN1086
C                                                                      *TRN1087
C               D O   F R E Q U E N C Y   B O O K K E E P I N G        *TRN1088
C                                                                      *TRN1089
C     ******************************************************************TRN1090
C                                                                       TRN1091
C               SET UP MAX FREQ BOUNDARY                                TRN1092
C                                                                       TRN1093
      HNUP=1.0E6                                                        TRN1094
      HNUP4=1.0E24                                                      TRN1095
      IF (MFTAG.EQ.0) GO TO 180                                         TRN1096
   60 IHNU=IHNU+1                                                       TRN1097
      CALL KAPPA                                                        TRN1098
      HNU4=HNU**4                                                       TRN1099
      DHNU=HNUP-HNU                                                     TRN1100
C                                                                       TRN1101
```

```
C     MERGE GROUPS WITH HNU MORE THAN (MERGE) TIMES LARGEST THETA       TRN1102
C                                                                       TRN1103
      IF (MERGE.GT.0.0) GO TO 70                                        TRN1104
      S1=7.0225                                                         TRN1105
      GO TO 965
   70 IF (THTAMX-HNU/MERGE) 100,80,80                                   TRN1107
   80 IF (IHNU-1) 90,210,120                                            TRN1108
   90 S1=7.0235                                                         TRN1109
      GO TO 965
C                                                                       TRN1111
C               REJECT TAPE IF MORE THAN HALF OF GROUPS MERGE           TRN1112
C                                                                       TRN1113
  100 IF (IHNU+IHNU-NHNU) 120,110,110                                   TRN1114
  110 Q=AMOD(MERGE,1.)                                                  TRN1115
      IF (Q.GT.0.4.AND.Q.LT.0.6) GO TO 120                              TRN1116
      S1=7.0250                                                         TRN1117
      GO TO 965
  120 DO 130 K=2,KMAX                                                   TRN1119
      IF (JMR(KFIT(K),2).NE.1) GO TO 130                                TRN1120
      T4=THETA(K)**4                                                    TRN1121
      BETA=HNU/THETA(K)                                                 TRN1122
      BETAP=HNUP/THETA(K)                                               TRN1123
      DFB=PLNKUT(BETA,BETAP)                                            TRN1124
      IF (DFB.LT.1.E-10) GO TO 130                                      TRN1125
      TEMP(1)=DFB*T4                                                    TRN1126
      EMB1=EXP(-BETA)                                                   TRN1127
      EMB2=EXP(-BETAP)                                                  TRN1128
      TEMP(2)=DFB+0.0384974/T4*(HNU4/(1.0-EMB1)*EMB1-HNUP4/(1.0-EMB2)*EMTRN1129
     1B2)                                                               TRN1130
C                                                                       TRN1131
C               FORM NUMERATORS AND DENOMINATORS OF MERGED KAPPAS       TRN1132
C                                                                       TRN1133
      XA(K)=XA(K)+TEMP(1)                                               TRN1134
      XB(K)=XB(K)+TEMP(2)                                               TRN1135
      B(K)=B(K)+PLANCK(K)*TEMP(1)                                       TRN1136
      ER(K)=ER(K)+TEMP(2)/ROSS(K)                                       TRN1137
  130 CONTINUE                                                          TRN1138
      HNUP=HNU                                                          TRN1139
      HNUP4=HNU4                                                        TRN1140
      IF (THTAMX-HNU/MERGE) 60,140,140                                  TRN1141
C                                                                       TRN1142
C               FORM MERGED KAPPAS                                      TRN1143
C                                                                       TRN1144
  140 DO 170 K=2,KMAX                                                   TRN1145
      IF (XA(K)) 150,170,160                                            TRN1146
  150 S1=7.0320                                                         TRN1147
      GO TO 965
  160 ROSS(K)=XB(K)/ER(K)                                               TRN1149
      PLANCK(K)=B(K)/XA(K)                                              TRN1150
      ER(K)=0.                                                          TRN1151
  170 CONTINUE                                                          TRN1152
      HNUP=1.0E6                                                        TRN1153
      HNUP4=1.0E24                                                      TRN1154
      DHNU=HNUP-HNU                                                     TRN1155
      GO TO 230                                                         TRN1156
```

```
C
C                  MONOFREQUENCY CALCULATION
C
  180 NHNU=1
      IHNU=1
      DO 190 K=2,KMAX
      IF (JMR(KFIT(K),2).NE.1) GO TO 190
      B(K) = 3.2732E+11 * THETA(K)*THETA(K)*THETA(K)*THETA(K)
C       3.2732E+11 = 1.0283E+12/PI    =   STEFANS  CONSTANT (EV)/PI
C       B = S/(4*PI*ROSS)
  190 CONTINUE
      HNU=.001
      GO TO 230
C     ************************************************************
C                                                                *
C               B E G I N   F R E Q U E N C Y   L O O P          *
C                                                                *
C     ************************************************************
  200 IHNU=IHNU+1
      CALL KAPPA
      DHNU=HNUP-HNU
      HNU4=HNU**4
C
C                  TYPICAL GROUP CALCULATION OF SOURCES
C
  210 DO 220 K=2,KMAX
      IF (JMR(KFIT(K),2).NE.1) GO TO 220
      DFB=PLNKUT(HNU/THETA(K),HNUP/THETA(K))
      B(K) = DFB * 3.2732E+11 * THETA(K)*THETA(K)*THETA(K)*THETA(K)
  220 CONTINUE
C
C                  SET BLACKBODY CONDITIONS
C
  230 IF (BCATAG .LT. 1.E-20)  GO TO 240
      DO 235 I=1,IMAX
      IF (ATHETA(I) .GT. 1.E-5) GO TO 234
      PA(I) = 0.
      GO TO 235
  234 PA(I) = PLNKUT(HNU/ATHETA(I),HNUP/ATHETA(I))
      PA(I) = PA(I)*TAU(I)/BETAA(I)*1.0283E12 * ATHETA(I)*ATHETA(I)
     1                                      * ATHETA(I)*ATHETA(I)
  235 CONTINUE
  240 IF (BCRTAG .LT. 1.E-20) GO TO 250
      DO 245 J=1,JMAX
      IF (RTHETA(J) .GT. 1.E-5) GO TO 244
      PRI(J) = 0.
      GO TO 245
  244 PRI(J) = PLNKUT(HNU/RTHETA(J),HNUP/RTHETA(J))
      PRI(J) = PRI(J)*PI*2.*X(IMAX)*DY(J)/BETAR(J) * 1.0283E12
     1  * RTHETA(J)*RTHETA(J)*RTHETA(J)*RTHETA(J)
  245 CONTINUE
  250 IF (BCBTAG .LT. 1.E-20) GO TO 270
      DO 255 I=1,IMAX
      IF (BTHETA(I) .GT. 1.E-5) GO TO 254
      PB(I) = 0.
```

```
      GO TO 255
 254  PB(I) = PLNKUT(HNU/BTHETA(I),HNUP/BTHETA(I))
      PB(I) = PB(I)*TAU(I)/BETAB(I)*1.0283E12 * BTHETA(I)*BTHETA(I)
     1                                       * BTHETA(I)*BTHETA(I)
 255  CONTINUE                                                              TRN1205
  270 CALL DVCHK (KDMY)                                                     TRN1206
      GO TO (280,290), KDMY                                                 TRN1207
  280 S1=7.0522
      GO TO 965
C                                                                           TRN1209
C                FORM ROSSELAND AND PLANCK OPTICAL DEPTHS                   TRN1210
C                                                                           TRN1211
C     DOUBLE ON STORAGE FOR ABSORPTION COEFFICIENTS, MU, AND LAMBDA         TRN1212
  290 DO 300 I=1,IMAX                                                       TRN1213
      W2(I)=0.                                                              TRN1214
      K=I+1                                                                 TRN1215
      M=K-IMAX                                                              TRN1216
      DO 300 J=1,JMAX                                                       TRN1217
      FACTOR=1.                                                             TRN1218
      IF (ABS(RPTAG).GT.1.E-20) FACTOR=ROSS(K)/PLANCK(K)                    TRN1219
      ROSS(K)=AMAX1(ROSS(K)*AMX(K)/(TAU(I)*DY(J)),1.E-20)
      PLANCK(K)=AMAX1(PLANCK(K)*FACTOR*AMX(K)/(TAU(I)*DY(J)),1.E-20)
C     USE ONLY ROSS FOR NOW.  ALAMH AND ALAMV USED FOR RAY COUNT            TRN1222
C     USING PLANCK AS ENERGY FLUX STORAGE IN RAY LOOP FOR DEBUG AND W2 ETRN1223
                                                                            TRN1224
      M=N+IMAX                                                              TRN1225
      K=K+IMAX                                                              TRN1226
  300 CONTINUE
C     *****************************************************************TRN1227
C                                                                      *TRN1228
C     T W O - D I M E N S I O N A L   T R A N S P O R T   S E C T I O N*TRN1229
C                                                                      *TRN1230
C     *****************************************************************TRN1231
      DO 306 J=1,JMAX
      DO 305 I=1,IMAX
      K = I + 1 + IMAX*(J-1)
      ALAMV(K) = 0.
      IF ( CLAMDA/ROSS(K) .LT. AMIN1(DY(J),DX(I))   .AND.
     1      ( JMR(KFIT(K),2).EQ.1)     )  ALAMV(K)=1.
 305  CONTINUE
 306  CONTINUE
C   ALAMV(K)=1. IMPLIES CELL K IS OPTICALLY THICK AND ACTIVE
      REWIND 3                                                              TRN1232
C     KXB, XB IS -BUFFER- STORAGE                                           TRN1233
C     KXA, XA USED FOR BOTH CELL INDEX AND PATH LENGTH(TRAN2 ONLY)          TRN1234
C
C   R A Y - T A P E   R E C O R D   L O O P   (NGRP RAYS)
C
 310  READ (3)  NGWD,(XB(I),I=2,NGWD)
      NGRP=KXB(2)                                                           TRN1236
      IGWD=2                                                                TRN1237
      IF (IABS(NGRP).LE.210) GO TO 320                                      TRN1238
      S1=7.0541                                                             TRN1239
      GO TO 965
C
C   R A Y   L O O P
```

```
C                                                                   TRN1241
  320 DO 820 IGRP=1,NGRP                                            TRN1242
      ITOT=KXB(IGWD+1)                                              TRN1243
      NTER=KXB(IGWD+2)                                              TRN1244
      NAIT=KXB(IGWD+3)
      WT = XB(IGWD+4)
      II = IGWD + 4                                                 TRN1246
      DO 330 I=1,ITOT                                               TRN1247
      KXA(I)=KXB(II+1)                                              TRN1248
      XA(I+400)=XB(II+2)                                            TRN1249
  330 II=II+2                                                       TRN1250
      IGWD=II                                                       TRN1251
C     TEST FOR -ALL RAYS PROCESSED- SIGNAL                          TRN1252
      IF (ITOT.EQ.1.AND.KXA(1).LT.0) GO TO 830                      TRN1253
C     BYPASS IF RAY TRAVERSES TOTALLY INACTIVE REGION               TRN1254
      IACT=1                                                        TRN1255
      DO 340 I=1,ITOT                                               TRN1256
      KT=KXA(I)                                                     TRN1257
      IF (JHR(KFIT(KT),2).NE.1) GO TO 340                           TRN1258
      IACT=2                                                        TRN1259
  340 CONTINUE                                                      TRN1260
      GO TO (720,350), IACT                                         TRN1261
C     GO FORWARD                                                    TRN1262
  350 IF (NTER.GE.1.AND.NTER.LE.3) GO TO 360                        TRN1263
      S1=7.0550
      GO TO 965                                                     TRN1265
  360 KSTRT=KXA(1)                                                  TRN1266
      IPASS=1                                                       TRN1267
C     APPLY BOUNDARY CONDITION TO START TRACE OF RAY                TRN1268
      GO TO (370,400,420), NTER                                     TRN1269
  370 IF (BCBTAG) 380,440,390                                       TRN1270
  380 S1=7.0556
      GO TO 965                                                     TRN1272
  390 I=MOD(KSTRT-2,IMAX)+1
      AJ = PB(I) * WT                                               TRN1275
      GO TO 450                                                     TRN1276
  400 IF (BCRTAG) 380,440,410                                       TRN1277
  410 J=(KSTRT-2)/IMAX+1
      AJ = PR(J) * WT                                               TRN1280
      GO TO 450                                                     TRN1281
  420 IF (BCATAG) 380,440,430
  430 I = MOD(KSTRT-2,IMAX) + 1
      AJ = PA(I) * WT                                               TRN1285
      GO TO 450                                                     TRN1286
  440 AJ=0.                                                         TRN1287
C     BEGIN LOOP OVER ZONES PASSED THROUGH BY RAY                   TRN1288
C     FOR DEBUG ONLY, USE PLANCK ARRAY TO STORE AJ                  TRN1289
C     XALP IS EFFECTIVE AREA OF RAY IN ZONE
  450 PLANCK(1) = AJ
      KOLD = 1
      ALAMV(KOLD) = 0.
C   OUTSIDE OF SYSTEM IS CONSIDERED OPTICALLY THIN
C
C   S E G M E N T   L O O P
C
```

```
      DO 540 IDY=1,ITOT                                                 TRN1297
      GO TO (490,500), IPASS                                            TRN1298
  490 ITR=IDY                                                           TRN1299
      GO TO 510                                                         TRN1301
  500 ITR=ITOT+1-IDY                                                    TRN1302
  510 K=KXA(ITR)                                                        TRN1304
 520  IF (ALAMV(KOLD) .GT. 0.5) GO TO 525
C
C   PREVIOUS ZONE WAS THIN
C
      IF (ALAMV(K) .LT. 0.5) GO TO 528
C   PRESENT ZONE IS THICK
      ER(K) = ER(K) + AJ
      AJN = AJ
      IF (IDY.LT.ITOT) GO TO 530
      AJN = B(K)*XALP(K) *WT
      ER(K) = ER(K) - AJN
      GO TO 530
C
C   PREVIOUS ZONE WAS THICK
C
 525  IF (ALAMV(K) .LT. 0.5) GO TO 527
C   PRESENT ZONE IS THICK
      IF (IDY .LT. ITOT) GO TO 529
      AJN = B(K)*XALP(K)*WT
      ER(K) = ER(K) - AJN
      GO TO 530
C   PRESENT ZONE IS THIN
 527  AJ = B(KOLD)*XALP(KOLD)*WT
      ER(KOLD) = ER(KOLD) - AJ
 528  IF (JMR(KFIT(K),2).EQ.0) GO TO 529
C   NORMAL TRANSPORT CALCULATION
      SIGS = ROSS(K)*XA(ITR+400)
      ESS = EXP(-SIGS)
      AJN = AJ * ESS + XALP(K) * B(K) * (1.-ESS) * WT
      ER(K) = ER(K) + AJ - AJN
C           ER = ERGS/SEC          XALP = EFFECTIVE AREA
      GO TO 530
C   PRESENT ZONE IS INACTIVE OR THICK
 529  AJN = AJ
 530  PLANCK(IDY+1) = AJN
      KOLD = K
  540 AJ=AJN                                                            TRN1317
C   IF PLANCK DEBUG IS DESIRED IT SHOULD BE ADDED HERE
C   ON SECOND PASS ORDER OF PLANCK PRINT WILL BE SAME AS TRACE DIRECTION
C   F I N I S H   S E G M E N T   L O O P
C     HANGE ETH BY ENERGY ENTERING AND LEAVING MESH                     TRN1322
 550  FOO = FOO + (PLANCK(1) - PLANCK(ITOT+1))
C     EVALUATE CONTRIBUTION TO FLUX ACROSS TOP BOUNDARY                 TRN1324
      IF (BCATAG.LT.0.) GO TO 610                                       TRN1325
      IF (INTER.NE.3) GO TO 580                                         TRN1326
      I=MOD(KXA(1)-2,IMAX)+1                                            TRN1327
C     IN ERGS/CM**2/SEC
      GO TO (560,570), IPASS                                            TRN1329
  560 W2(I)=W2(I)-PLANCK(I)/TAU(I)
```

```
      GO TO 580                                                          TRN1331
  570 W2(I)=W2(I)+PLANCK(ITOT+1)/TAU(I)
  580 IF (NAI).NE.3) GO TO 610                                           TRN1333
      I=MOD(KXA(ITOT)-2,IMAX)+1                                          TRN1334
C     IN ERGS/CM**2/SEC
      GO TO (590,600), IPASS                                             TRN1336
  590 W2(I)=W2(I)+PLANCK(ITOT+1)/TAU(I)
      GO TO 610                                                          TRN1338
  600 W2(I)=W2(I)-PLANCK(I)/TAU(I)
  610 GO TO (620,720), IPASS                                             TRN1340
  620 CONTINUE
  630 KSTRT=KXA(ITOT)                                                    TRN1345
      GO TO (640,650,660), NTER                                          TRN1346
  640 EBTM=EBTM+PLANCK(I)                                                TRN1347
      GO TO 670                                                          TRN1348
  650 ESIDE=ESIDE+PLANCK(I)                                              TRN1349
      GO TO 670                                                          TRN1350
  660 ETOP=ETOP+PLANCK(I)                                                TRN1351
  670 IF (NXIT.GT.0)   GO TO (680,690,700), NXIT
      GO TO (680,690,700) , NTER
C   THETA=0 CASE AND REFLECTION -- RAY ENTERS AND EXITS THRU BOTTOM
  680 EBTM=EBTM-PLANCK(ITOT+1)                                           TRN1353
      GO TO 710                                                          TRN1354
  690 ESIDE=ESIDE-PLANCK(ITOT+1)                                         TRN1355
      GO TO 710                                                          TRN1356
  700 ETOP=ETOP-PLANCK(ITOT+1)                                           TRN1357
  710 IPASS=2                                                            TRN1358
C   IF THETA=0 AND REFLECTIVE BOUNDARY, ONLY MAKE ONE PASS ALONG RAYS
      IF (NXIT.LE.0) GO TO 800
      GO TO (370,400,420), NXIT                                          TRN1359
C     BOTH PASSES COMPLETE -- RETURN FOR NEXT RAY                        TRN1360
  720 GO TO (730,740,750), NTER                                          TRN1361
  730  EBTM = EBTM - PLANCK(ITOT+1)
       GO TO 760                                                         TRN1363
  740  ESIDE = ESIDE - PLANCK(ITOT+1)
       GO TO 760                                                         TRN1365
  750  ETOP = ETOP - PLANCK(ITOT+1)
  760 GO TO (770,780,790), NXIT                                          TRN1367
  770  EBTM = EBTM + PLANCK(I)
       GO TO 800                                                         TRN1369
  780  ESIDE = ESIDE + PLANCK(I)
       GO TO 800                                                         TRN1371
  790  ETOP = ETOP + PLANCK(I)
  800 CALL UVCHK (KDMT)                                                  TRN1373
      GO TO (810,820), KDMY                                              TRN1374
  810 S1=7.0810                                                          TRN1375
      GO TO 965
  820 CONTINUE                                                           TRN1377
C   E N D   R A Y   L O O P
      GO TO 310                                                          TRN1378
C
C   A L L   R A Y S   T R A C E D — T R A N S P O R T   C O M P L E T E
C
C   START OF THICK-THICK EXPLICIT DIFFUSION CALCULATION
C
```

```
C  8/3PI=8.37758  16/3PI**2=52.63789   THIS EXPLICIT DIFFUSION
C              CALCULATION IS ONLY VALID FOR MONOFREQUENCY CASE
 830  DO 825 I=1,IMAX
      KPP = I+1
      J = 0
      AV = TAU(I) * 8.37758
      AH = X(I) * 52.63789
 821  K = KPP
      KPP = K + IMAX
      J = J + 1
      IF (J.GT.JMAX) GO TO 825
      IF (ALAMV(K) .LT. 0.5) GO TO 821
C   CELL K IS THICK AND ACTIVE
      IF ( (I.GE.IMAX) .OR. (ALAMV(K+1).LT.0.5) )  GO TO 822
C   CALCULATE NET DIFFUSION IN HORIZONTAL DIRECTION BETWEEN CELLS K,K+I
      DERH = DY(J)*(B(K)-B(K+1))/(ROSS(K)*DX(I) + ROSS(K+I)*DX(I+1))*AH
      ER(K) = ER(K) - DERH
      ER(K+1) = ER(K+1) + DERH
 822  IF ( (J.GE.JMAX) .OR.(ALAMV(KPP).LT.0.5) ) GO TO 821
C   CALCULATE NET DIFFUSION IN VERTICAL DIRECTION BETWEEN CELLS K,KPP
      DERV = (B(K)-B(KPP)) / (ROSS(K)*DY(J) + ROSS(KPP)*DY(J+I)) * AV
      ER(K) = ER(K) - DERV
      ER(KPP) = ER(KPP) + DERV
      GO TO 821
 825  CONTINUE
C   FINISH OF THICK-THICK EXPLICIT DIFFUSION CALCULATION
C
C
      REWIND 3
      HNUP=HNU                                                          TRN1381
      HNUP4=HNU4                                                        TRN1382
      IF (IHNU-NHNU) 200,850,840                                        TRN1383
  840 S1=7.1010                                                         TRN1384
      GO TO 965
  850 CONTINUE
  870 WRITE (6,970) NC                                                  TRN1390
      IF (ABS(DBGPRT) .LT. 1.E-20) GO TO 880
      WRITE (6,980) (ER(K),K=2,KMAX)                                    TRN1391
      ESUM = 0.
      DO 331 K=2,KMAX
      ESUM = ESUM + ER(K)
      ALAMV(K) = B(K) * 4. * PI * ROSS(K)
  331 ALAMH(K)=ER(K)*(TAU(I)*DY(J)*ROSS(K))+4.*PI*B(K)
      WRITE (6,9010) (ALAMH(K),K=2,KMAX),ESUM
      WRITE (6,9010) (ROSS(K),K=2,KMAX)
      WRITE (6,9010) (B(K),K=2,KMAX)
      WRITE (6,9010)  (ALAMV(K),K=2,KMAX)
C                                                                       TRN1392
C     ADVANCE FREQ, STORE EMERGENT FLUX, TEST FOR COMPLETION OF GROUPS  TRN1393
C                                                                       TRN1394
C     *****************************************************************TRN1395
C                                                                      *TRN1396
C               E N D   F R E Q U E N C Y   L O O P                    *TRN1397
C                                                                      *TRN1398
C     *****************************************************************TRN1399
```

```
C     ITERATE ON TEMPERATURE                                            TRN1400
  880 GO TO (890,910), NVEZ                                             TRN1401
  890 NVEZ=2                                                            TRN1402
      VEZ=NVEZ                                                          TRN1403
      NY=NVEZ                                                           TRN1404
      ISNU=0                                                            TRN1405
      DO 900 K=2,KMAX                                                   TRN1406
C     WORK, SOURCE TERMS OMITTED                                        TRN1407
      IF (JMR(KFIT(K),2).NE.1) GO TO 900                                TRN1408
      OLDTH(K)=THETA(K)                                                 TRN1409
      E=AIX(K)+ER(K)*DT/AMX(K)                                          TRN1410
      SV=TAD(I)*DY(J)/AMX(K)
      CALL ES (SV,E,TEMP(1),TEMP(2),GG)                                 TRN1412
      THETA(K)=0.5*(THETA(K)+TEMP(1))                                   TRN1413
  900 CONTINUE                                                          TRN1414
      IF (MFTAG.EQ.0) CALL KAPPA                                        TRN1415
      GO TO 40                                                          TRN1416
  910 IF (ITAG.EQ.0) GO TO 930                                          TRN1417
      DO 920 K=2,KMAX                                                   TRN1418
  920 THETA(K)=OLDTH(K)                                                 TRN1419
C     H A N G E   I N T E R N A L   E N E R G I E S                     TRN1420
  930 TEMP(1)=1.                                                        TRN1421
      DO 940 K=2,KMAX                                                   TRN1422
      IF (JMR(KFIT(K),2).NE.1) GO TO 940                                TRN1423
      DE=ER(K)*DT/AMX(K)                                                TRN1424
      Q=SLUG*AIX(K)                                                     TRN1425
      IF(ABS(DE).LT.ABS(Q)) GO TO 940
      TEMP(1)=AMIN1(Q/ABS(DE),TEMP(1))                                  TRN1427
  940 CONTINUE                                                          TRN1428
      DTEMP=DT*TEMP(1)
      IF(DTEMP.GT.FFB) GO TO 950
      WRITE (6,970) NC                                                  TRN1431
      WRITE (6,980) (ER(K),K=2,KMAX)                                    TRN1432
      WRITE (6,990) DE,K,AIX(K),AMX(K)                                  TRN1433
      S1=7.1068                                                         TRN1434
      GO TO 965
  950 T=T-DT+DTEMP
      DT=DTEMP
      FOO=FOO*DT
      ETH = ETH + FOO
      BACC = BACC + EBTM*DT
      SACC = SACC + ESIDE*DT
      TACC = TACC + ETOP*DT
C   FOO,ETH,BACC,SACC,TACC ARE IN ERGS
      WRITE (6,1000) NC,DT                                              TRN1447
      WRITE (6,1010)  BACC,SACC,TACC,FOO
      READ(4) FIOUT,P,U,V
      RETURN                                                            TRN1451
 967  S1 = 7.1111
  965 READ(4) FIOUT,P,U,V
      ISEND = 2
      CALL EDIT
C                                                                       TRN1452
 970  FORMAT (52HUINTERNAL ENERGY GAIN PER UNIT TIME DUE TO RADIATION,8H
     1 CYCLE =I5)                                                       TRN1454
```

```
 980 FORMAT (1X,1P10E11.4)                                              TRN1455
 990 FORMAT (5H DE =1PE11.4,5H  K =I10,6H AIX =E11.4,6H AMX =E11.4)     TRN1456
1000 FORMAT (19H0   FLUX FOR CYCLE I4,7H   DT =1PE10.3)                 TRN1457
1010 FORMAT (9H BOTTOM =1PE11.4,8H  SIDE =E11.4,7H  TOP =E11.4,7H  FOO TRN1458
    1=E11.4)                                                            TRN1459
9010 FORMAT (1H0/(1X,1P10E11.4))
     END                                                                TRN1460
```

SHORT2 ROUTINES

DRAW Subroutine Sequence
- Subroutine DRAW
- Subroutine INWARD
- Subroutine OUTWRD
- Subroutine DISTNC
- Function JO(J, I, L, M)
- Function JB(J, I, L, M)
- Function SINT(A1, B1, EPS1, FUNC)
- Function IN(Z)
- Function BOT(Z)
- Function OUT(Z)
- Function GAMMO(IG, Z)

Transport Subroutine Sequence
- Subroutine TRAN2
- Function TRANS(SIGMA, D, XA, XB, YA, YB, C, SUM, IACT, SP)
- Function S(XX, YY)
- Function SRC(KKK)
- Subroutine BOTBND
- Subroutine OUTBND
- Subroutine TOPBND

This page intentionally left blank.

Subroutine DRAW

```
C     DRAW
      SUBROUTINE DRAW                                                    DRA8001
C                                                                        DRA8006
C     ******************************************************************DRA8007
C          H E C T I C   C O M M O N                                     DRA8008
C     ******************************************************************DRA8009
C                                                                        DRA8010
      COMMON AHN,    AMBIEN, ANN,    ATOM,   BCATAG, BCBTAG, BCLTAG,     HECCOM01
     1BCRTAG, CAPIN, CAPS,   COUT,   CMXK,   CMXOM,  CNDE,   CO,         HECCOM02
     2COE,    CSTOP, CCVV,   CYCLE,  DBGPRT, DGREY,  DHNU,   DIANTP,     HECCOM03
     3DMIN,   DNN,   DT,     DTBUGR, DTC,    DTH,    DTNA,   DTQ,        HECCOM04
     4DTH,    DTR1,  DTR2,   DTUF,   DTVF,   DUMPT7, ECK,    EFRAC,      HECCOM05
     5EI1,    ERRCRT, ETH,   EZERO,  FDIC,   FFA,    FFB,    FIFT,       HECCOM06
     6FUDGE,  GV,    HCB,    HCP,    HH,     HNU,    HNUP,   HNUS,       HECCOM07
     7HVB,    I,     IC,     IGOTO,  IH,     IHNU,   IMAX,   IMPTAG,     HECCOM08
     8ISSSS,  ISR,   ITAG,   ITRMAX, IU,     IV,     I1,     I2,         HECCOM09
     9J,      JC,    JH,     JMAX,   JMAXA,  JU,     JV,     J5          HECCOM10
      COMMON K,      KMAXA,  KMN1,   KMN2,   KP,     LLLL,   MMMM,       HECCOM11
     1MERGE,  MFTAG, M2,     N,      NC,     NHNU,   NK,     NPC,        HECCOM12
     2NR,     NRM,   NT,     NTAPE,  NY,     N1,     N2,     N3,         HECCOM13
     3N4,     ODDC,  PABOVE, PBLO,   PI,     PIDTS,  PRINTL, PRINTS,     HECCOM14
     4PROB,   PRR,   PW,     RADE,   RC,     RFT,    RPTAG,  RR,         HECCOM15
     5SCDR,   SCR,   SCHE,   SCYCLE, SGNL,   SIG,    SIGH,   SLUG,       HECCOM16
     6SN,     SPROB, SVMAX,  SVS,    S1,     S2,     S3,     S4,         HECCOM17
     7SS,     T,     TAUDTS, THICK,  TMAX,   TRAD,   URR,    UT,         HECCOM18
     8UU,     UVMAX, VABOVE, VAPE,   VBLO,   VEL,    WS,     WSA,        HECCOM19
     9XMAXX,         BLANK(49)                                           HECCOM20
      COMMON         AIX(1200),      AMX(1200),      ALAMH(1200),
     1ALAMV(1200),   B(1200),        CAP(1200),      ER(1200),
     2FIOUT(1200),   KFIT(1200),     P(1200),        SMLG(1200),
     3THETA(1200),   U(1200),        V(1200),        XALP(1200),
     4XB(1200),      SOLID(400),     PL(200),        Y(101),
     5DY(100),       GAMC(100),      YAMC(100),      UL(100),            HECCOM26
     6W2(100),       X(53),          DX(52),         TAU(52),            HECCOM27
     7BETAB(50),     TEMP(12),       HEAD(12)                            HECCOM28
C                                                                        DRA8038
C    THE FOLLOWING VARIABLES IN HECTIC COMMON HAVE BEEN RENAMED IN THIS
C    ROUTINE  --  CV TO CCVV , ISEND TO ISSSS , L TO LLLL , M TO MMMM,
C    AND XMAX TO XMAXX
C     ******************************************************************DRA8039
C                                                                        DRA8040
      DIMENSION      FLEFT(100),     PR(100),        SIGC(100)           DRA8041
      EQUIVALENCE    (FLEFT,UL),  (PL,PR)
      EQUIVALENCE    (PL(101),SIGC)
      EQUIVALENCE (S20,BLANK(12)), (ZP126,BLANK(13)), (ZP136,BLANK(14)),
     1(KMX,BLANK(15)),(WT,BLANK(16)),(ANUMBR,BLANK(19)),(LOCZ,BLANK(22))
     2,(H,BLANK(25)),(NA,BLANK(26))
      EQUIVALENCE  (BACC,BLANK(47)),(SACC,BLANK(48)),(TACC,BLANK(49))
C                                                                        DRA8045
      REAL  MERGE                                                        DRA8046
C                                                                        DRA8047
C                                                                        DRA8048
      COMMON     /RAD1/    IAID,     KMAX,     ICHECK,   ISEND,          RAD1
     1L2,        L2P1,     LMAX,     M2,       MUMAX,    ISENDD,         RAD2
     2ISWELP,    LLOW,     LMI                                           RAD
      COMMON     /RAD2/    ANGL1,    ANGL2,    ASN,      CHKSUM,         RAD3
```

```
     1DELTY,    FORPI,    HELP1,    HELP2,    LMDA,    PP,             RAD4
     2PI2,      PI4,      SINP2,    TANN,     TOLER,   XMIN,           RAD5
     3XMAX,     STEF                                                   RAD6
      COMMON    /RAD3/              DUMI(1),           AA(16),         RAD7
     1          COSPHI(6),          CVIN(6),           CVOUT(6),       RAD8
     2          CH(6),              MU(6),             PHI(6),         RAD9
     3          DUM2(1),            SINLP(6),          SMU(6),         RAD10
     4          W(6),               CV(6)                              RAD11
      COMMON    /INDEX/   M,L,LP                                       RAD12
      COMMON    /DRUMM/   LCHECK,   KK2,      NOBUFF                   RAD13
      COMMON    /DIS/     DV(6),    XV(6),    YV(6),    DH(6),         RAD14
     1          XH(6),    YH(6)                                        RAD15
      COMMON    /SUMCHK/            SUMB(50,6),        SUMO(50,6)      RAD16
      COMMON    /BNDT/    BTHETA(50),     RTHETA(50),    ATHETA(50)    RAD17
C
      REAL     MU , LMDA
C
      EQUIVALENCE   (FIOUT,OLDTH) , (U,ROSS) ,
     .  (CAP(1),BUFF1(1),BUFF(1),NOI) , (P(1),PLANCK(1),BUFF2(1),NOO)
      DIMENSION   OLDTH(1) , ROSS(I) , BUFF1(1) , BUFF(1) , PLANCK(1) ,
     .  BUFF2(1)
C
      EQUIVALENCE   (V(1),XA(1),EB(1))
      DIMENSION   XA(1)
      DIMENSION             EB(50,6) , EV(50,6) , ET(50,6) ,
     .          ETAIO(6,6) , ETAIT(6,6) , ETABI(6,6) , ETABT(6,6),
     .  ETABO(6,6) , ETAOI(6,6) , ETAOT(6,6) , ETAOO(6,6)
C
C
C
      REWIND   4
      WRITE (4)  FIOUT,CAP,P,U,V
      REWIND   4
      REWIND   3
      KMAX = KMAXA-1
      STEF = 1.0283E+12
      PI2 = PI/2.
      FORPI = 4.*PI
      PI4 = PI/4.
      READ (5,9500)  LMAX,MUMAX,TOLER,CHKSUM
      WRITE (6,9510)  LMAX,MUMAX,TOLER,CHKSUM
      READ (5,9600)  (ATHETA(I),I=1,IMAX)
      READ (5,9600)  (RTHETA(J),J=I,JMAX)
      READ (5,9600)  (BTHETA(I),I=1,IMAX)
      WRITE (6,9702)  (ATHETA(I),I=I,IMAX)
      WRITE (6,9701)  (RTHETA(J),J=1,JMAX)
      WRITE (6,9700)  (BTHETA(I),I=1,IMAX)
      PP = PI/FLOAT(LMAX)
      L2 = LMAX/2
      L2PI = L2 + 1
      DO 270 L=1,LMAX
      PHI(L) = PP * (FLOAT(L) - .5)
  270 COSPHI(L) = COS(PHI(L))
      DO 275 L=0,LMAX
  275 SINLP(L) = SIN(PP*FLOAT(L))
```

```
      SINP2 = SIN(PP/2.)
      M2 = MUMAX/2
C     MU=1 AND MU=0 ARE PROHIBITED CASES
C        REQUIREMENT --
C    CHOICE OF MUS MUST BE SYMMETRIC ABOUT 0.
      DO 280 L=1,LMAX
      MU(L) = COSPHI(L)
 280  SMU(L)= SIN(PHI(L))
      DO 600 M=1,LMAX
 600  W(M) = ABS( COS(FLOAT(M)*PP) - COS(FLOAT(M-1)*PP) ) * .5
C
C
C
      KK2 = 1
      NOO = 0
      DO 3000 M=1,M2
      DO 2000 J=1,JMAX
      NOBUFF = 1
C    ZERO OUT SUMCHECKS
      DO 650 K=1,LMAX
      DO 650 N=1,IMAX
      SUMB(N,K) = 0.
 650  SUMO(N,K) = 0.
      ICHECK = 0
C    GOING INWARD
      I = IMAX
 700  CALL INWARD
      I = I-1
      IF (I .GE. 1)  GO TO 700
C    GOING OUTWARD
      DO 1000 I=1,IMAX
      CALL  OUTWRD
 1000 CONTINUE
C    SUMCHECK
      IF (ICHECK .EQ. 0)  WRITE (6,9054)  M,J
      IBB = 0
      IOO = 0
      DO 1500 N=1,IMAX
      DO 1500 K=1,LMAX
       IF (ABS(SUMB(N,K)-1.)  .LT.  CHKSUM)  GO TO 1450
      WRITE (6,9050)  M,J,N,K,SUMB(N,K)
      IBB = 1
 1450 CONTINUE
      IF ( ABS(SUMO(N,K)-1.).LT.CHKSUM  .OR.  K.GT.L2)  GO TO 1499
      WRITE (6,9051)  M,J,N,K,SUMO(N,K)
      IOO = 1
 1499 CONTINUE
 1500 CONTINUE
      IF (IBB .EQ. 0)  WRITE (6,9052)  M,J
      IF (IOO .EQ. 0)  WRITE (6,9053)  M,J
 4000 WRITE (3)  (BUFF2(ILM),ILM=1,1023)
      KK2 = 1
      NOO = 0
 2000 CONTINUE
 3000 CONTINUE
```

```
C
C
      WRITE (3)  (BUFF2(ILM),ILM=1,1023)
      READ (4)  FIOUT,CAP,P,U,V
      REWIND   4
      WRITE (6,9876)  NOBUFF
      RETURN
C
 9050 FORMAT (7H0FOR M=,I2,3H J=,I2,3H I=,I2,4H LP=,I2,12H SUMB(I,LP)=,1
     .PE14.6)
 9051 FORMAT (7H0FOR M=,I2,3H J=,I2,3H I=,I2,4H LP=,I2,12H SUMO(I,LP)=,1
     .PE14.6)
 9052 FORMAT (7H0FOR M=,I2,3H J=,I2,29H CHECKSUMS FOR BOTTOM ETAS=1.)
 9053 FORMAT (7H0FOR M=,I2,3H J=,I2,30H CHECKSUMS FOR OUTSIDE ETAS=1.)
 9054 FORMAT (7H0FOR M=,I2,3H J=,I2,29H CHECKSUMS FOR INSIDE ETAS=1.)
 9500 FORMAT (2I6,2E12.5)
 9510 FORMAT (12H1 DRAW INPUT/5X,5HLMAX=,I6/5X,6HNUMAX=,I6/5X,6HTOLER=,1
     .PE14.6/5X,7HCHKSUM=,1PE14.6///)
 9600 FORMAT (6E12.5)
 9700 FORMAT (1H0,10X,9HATHETA(I)/(1P8E14.6))
 9701 FORMAT (1H0,10X,9HRTHETA(J)/(1P8E14.6))
 9702 FORMAT (1H0,10X,9HATHETA(I)/(1P8E14.6))
 9876 FORMAT (31H1 NUMBER OF BUFFERS PER LAYER =,I5///)
      END
```

Subroutine INWARD

```
      SUBROUTINE  INWARD
C     GIVEN INDICES  M,J,I  COMPUTE, PRINT, AND SAVE
C        1)  ETABI(L,LP)                  LP.LE.L.LE.L2
C        2)  ETAOI(L,LP)                  LP.LE.L.LE.L2
C        3)  ETABT(L,LP)                  LP.LE.L.LE.L2  (EXITING INWARD)
C        4)  ETAOT(L,LP)                  LP.LE.L.LE.L2  (EXITING INWARD)
C        5)  DV(L),XV(L),CV(L),DH(L),XH(L),CH(L)
C
C     ************************************************************DRA8007
C         H E C T I C   C O M M O N                              DRA8008
C     ************************************************************DRA8009
      COMMON  AHN,    AMBIEN, ANN,    ATOM,   BCATAG, BCBTAG, BCLTAG,   HECCOM01
     1BCRTAG, CAPIN,  CAPS,   CDUT,   CMXK,   CM[illegible], CNDE,   CO,       HECCOM02
     2CDE,    CSTOP,  CCVV,   CYCLE,  DBGPRT, DGHET,  DHNU,   DIANTP,   HECCOM03
     3DMIN,   DNN,    DT,     DTBUGR, DTC,    DTH,    DTNA,   DTQ,      HECCOM04
     4DTR,    DTR1,   DTR2,   DTUF,   DTVF,   DUMPT7, ECK,    EFRAC,    HECCOM05
     5E11,    EHRCRT, ETH,    EZERO,  FOIC,   FFA,    FFB,    FIFT,     HECCOM06
     6FUDGE,  GV,     HCB,    HCP,    HH,     HNU,    HNUP,   HNUS,     HECCOM07
     7HVB,    I,      IC,     IGOTO,  IH,     IHNU,   IMAX,   IMPTAG,   HECCOM08
     8ISSSS,  ISR,    ITAG,   ITRMAX, IU,     IV,     I1,     I2,       HECCOM09
     9J,      JC,     JH,     JMAX,   JMAXA,  JU,     JV,     J5        HECCOM10
      COMMON  K,      KMAXA,  KMN1,   KMN2,   KP,     LLLL,   MMMM,     HECCOM11
     1MERGE,  MFTAG,  MZ,     N,      NC,     NHNU,   NK,     NPC,      HECCOM12
     2NR,     NRM,    NT,     NTAPE,  NT,     N1,     N2,     N3,       HECCOM13
     3N4,     ODDC,   PABOVE, PBLO,   P1,     PIDTS,  PRINTL, PRINTS,   HECCOM14
     4PROR,   PRR,    PW,     RADE,   RC,     RFT,    RPTAG,  RR,       HECCOM15
     5SCDR,   SCR,    SCHE,   SCTCLE, SGNL,   SIG,    SIGH,   SLUG,     HECCOM16
     6SN,     SPROB,  SVMAX,  SVS,    S1,     S2,     S3,     S4,       HECCOM17
     7SS,     T,      TAUDTS, THICK,  TMAX,   TRAD,   URR,    UT,       HECCOM18
     8UU,     UVMAX,  VABOVE, VAPE,   VBLO,   VEL,    WS,     WSA,      HECCOM19
     9XMAXX,          BLANK(49)                                         HECCOM20
      COMMON          AIX(1200),      AMX(1200),      ALAMH(1200),
     1ALAMV(1200),    B(1200),        CAP(1200),      ER(1200),
     2FIOUT(1200),    KFIT(1200),     P(1200),        SMLQ(1200),
     3THETA(1200),    U(1200),        V(1200),        XALP(1200),
     4XB(1200),       SOLID(400),     PL(200),        Y(101),
     5DT(100),        GAMC(100),      TAMC(100),      UL(100),          HECCOM26
     6W2(100),        X(53),          UX(52),         TAU(52),          HECCOM27
     7BETAB(50),      TEMP(12),       HEAU(12)                          HECCOM28
C                                                                       DRA8038
C     THE FOLLOWING VARIABLES IN HECTIC COMMON HAVE BEEN RENAMED IN THIS
C     ROUTINE  --  CV TO CCVV , ISEND TO ISSSS , L TO LLLL , M TO MMMM,
C     AND XMAX TO XMAXX
C     ************************************************************DRA8039
C                                                                       DRA8040
      DIMENSION       FLEFT(100),     PR(100),        SIGC(100)         DRA8041
      EQUIVALENCE     (FLEFT,UL),     (PL,PR)
      EQUIVALENCE     (PL(101),SIGC)
      EQUIVALENCE (S20,BLANK(12)), (ZP120,BLANK(13)), (ZF136,BLANK(14)),
     1(KMX,BLANK(15)),(WT,BLANK(16)),(ANUMBR,BLANK(19)),(LC2,BLANK(22))
     2,(H,BLANK(25)),(NA,BLANK(26))
      EQUIVALENCE (BACC,BLANK(47)),(SACC,BLANK(48)),(TACC,BLANK(49))
C                                                                       DRA8045
      REAL   MERGE                                                      DRA8046
C                                                                       DRA8047
C                                                                       DRA8048
```

```
      COMMON    /RAD1/    IAID,     KMAX,     ICHECK,   ISEND,          RAD1
     1L2,       L2P1,     LMAX,     M2,       MUMAX,    ISENDD,         RAD2
     2ISWEEP,   LLOW,     LMI                                           RAD
      COMMON    /RAD2/    ANGL1,    ANGL2,    ASN,      CHKSUM,         RAD3
     1DELTY,    FORPI,    HELP1,    HELP2,    LMDA,     PP,             RAD4
     2PI2,      PI4,      SINP2,    TANN,     TOLER,    XMIN,           RAD5
     3XMAX,     STEF                                                    RAD6
      COMMON    /RAD3/              DUM1(1),            AA(16),         RAD7
     1          COSPHI(6),          CVIN(6),            CVOUT(6),       RAD8
     2          CH(6),              MU(6),              PHI(6),         RAD9
     3          DUM2(1),            SINLP(6),           SMU(6),         RAD10
     4          W(6),               CV(6)                               RAD11
      COMMON   /INDEX/   M,L,LP                                         RAD12
      COMMON    /DRUMM/   LCHECK,   KK2,      NOBUFF                    RAD13
      COMMON    /DIS/     UV(6),    XV(6),    TV(6),    DH(6),          RAD14
     1          XH(6),    YH(6)                                         RAD15
      COMMON    /SUMCHK/            SUMB(50,6),         SUMO(50,6)      RAD16
      COMMON    /BNDT/    BTHETA(50),         RTHETA(50),    ATHETA(50) RAD17
C
      REAL     MU , LMDA
C
      EQUIVALENCE   (FIOUT,OLDTH) , (U,ROSS) ,
     .   (CAP(1),BUFF1(1),BUFF(1),NOI) , (P(1),PLANCK(1),BUFF2(1),NOO)
      DIMENSION   OLDTH(1) , ROSS(1) , BUFF1(1) , BUFF(1) , PLANCK(1) ,
     .   BUFF2(1)
C
      EQUIVALENCE   (V(1),XA(1),EB(1))
      DIMENSION   XA(1)
      DIMENSION             EB(50,6) , EV(50,6) , ET(50,6) ,
     .          ETAIO(6,6) , ETAIT(6,6) , ETABI(6,6) , ETABT(6,6),
     .   ETABO(6,6) , ETAOI(6,6) , ETAOT(6,6) , ETAOO(6,6)
C
C
C
C
      REAL   LIM1 , LIM2
      REAL   JO , JB
      EXTERNAL    BOT
      EXTERNAL    OUT
C
C   C O M P U T E   S P E C I F I E D   C O N S T A N T S
C
      DO 40 L=0,LMAX
   40 AA(L) = SINLP(L) * X(I)
      DO 50 LP=1,L2
   50 CH(LP) = JB(J-1,I,LP,M)
      DO 60 LP=1,L2
      CVIN(LP) = JO(J,I-1,LP,M)
   60 CV(LP) = JO(J,I,LP,M)
      ISWEEP = 1
C
C   Z E R O   E T A S
      DO 500 L=1,L2
      DO 500 LP=1,L2
      ETABI(L,LP) = 0.
```

```
      ETAOI(L,LP) = 0.
      ETABT(L,LP) = 0.
      ETAOT(L,LP) = 0.
  500 CONTINUE
C
C   C O M P U T E   A N D   W R I T E   E T A B I
C   ETABI(L,LP)         = FRACTION OF (ENERGY IN FAN LP PASSING INTO CELL
C     (I,J) THRU THE BOTTOM OF THE CELL)  THAT PASSES OUT CELL (I,J)
C     THRU THE INSIDE WALL IN FAN L MEASURED THERE
      IF (I .NE. 1)  GO TO 510
      KK1 = 0
      GO TO 731
  510 ISEND = 1
      LMDA = DY(J) * (SMU(M)/MU(M))
      MELPI = DY(J)*DY(J)*(SMU(M)/MU(M))*(SMU(M)/MU(M)) - X(I-1)*X(I-1)
      LIM2 = AMIN1(X(I),X(I-1)+LMDA)
      XMIN = X(I-1)
      QP = 0.
      L = I
      LP = 1
  690 EPS = TOLER * CH(LP) / (FORPI*W(M)*ABS(MU(M)))
      ALPHAL = ASIN( X(I-1)*SINLF(L)/X(I) )
      IF ( ALPHAL .LT. PP*FLOAT(LP) )  GO TO 695
C   FAN LP CANNOT FILL FAN L AT X(I-1)
      ANGLI = X(I) * SINLP(LP)
      Q = SINT(X(I-1),LIM2,EPS,BOT)
      ETABI(L,LP) = (Q-QP) / CH(LP)
     .  * FORPI * W(M) * ABS(MU(M))
      LP = LP + 1
      QP = Q
      GO TO 690
C   FAN LP CAN FILL FAN L AT X(I-1)
  695 ANGLI = X(I-1) * SINLP(L)
      Q = SINT(X(I-1),LIM2,EPS,BOT)
      ETABI(L,LP) = (Q-QP) / CH(LP)
     .  * FORPI * W(M) * ABS(MU(M))
      QP = Q
      L = L + 1
      IF (L .LE. L2)  GO TO 690
C
      IF (DBGPRT .LT. 1.E-20)  GO TO 730
      WRITE (6,9000)  M,J,I
      DO 725 L=I,L2
      WRITE (6,9001)  L,(ETABI(L,LP),LP=I,L2)
  725 CONTINUE
  730 CONTINUE
C
C
C   C O M P U T E   A N D   W R I T E   E T A O I
C   ETAOI(L,LP)         = FRACTION OF (ENERGY IN FAN LP PASSING INTO CELL
C   (I,J) THRU THE OUTSIDE OF THE CELL) THAT PASSES OUT CELL(I,J) THRU
C   THE INSIDE IN FAN L
      ISEND = 1
      DELTY = DY(J)
      TANN = SMU(M)/MU(M)
```

```
      XMAX = X(I)
      XMIN = X(I-1)
      ASN = ASIN(X(I-1)/X(I))
      LIM2 = DY(J) - (X(I)-X(I-1))/TANN
      IF ( LIM2 .LT. 0. )  GO TO 1080
      L = 1
      LP = 1
      Q = 0.
 1055 EPS = TOLER * CV(LP) / (FORPI*X(I)*W(M)*SMU(M))
 1060 QP = Q
      IF ( X(I)*SINLP(LP) .GT. X(I-1)*SINLP(L) ) GO TO 1070
C   LP CANNOT FILL L
      ANGL1 = PP * FLOAT(LP)
      Q = SINT(0.,LIM2,EPS,OUT)
      ETAOI(L,LP) = (Q-QP) / CV(LP)
     .  * FORPI  * X(I) * W(M) * SMU(M)
      LP = LP + 1
      GO TO 1055
C   LP FILLS L
 1070 CONTINUE
      ANGL1 = ASIN( X(I-1)*SINLP(L)/X(I) )
      Q = SINT(0.,LIM2,EPS,OUT)
      ETAOI(L,LP) = (Q-QP) / CV(LP)
     .  * FORPI * X(I) * W(M) * SMU(M)
      L = L + 1
      IF (L .LE. L2) GO TO 1060
      GO TO 1100
 1080 DO 1090L=1,L2
      DO 1090 LP=1,L2
 1090 ETAOI(L,LP) = 0.
 1100 CONTINUE
C
      IF (DUGPRT .LT. 1.E-10)  GO TO 1120
      WRITE (6,9003)  M,J,I
      DO 1110 L=1,L2
      WRITE (6,9001)  L,(ETAOI(L,LP),LP=1,L2)
 1110 CONTINUE
 1120 CONTINUE
C   COMPUTE  DV(L),XV(L),YV(L)
      LLOW = L2P1
      LH1 = LMAX
      ISENDD = 1
      CALL  DISTNC
 1130 KK1 = 0
      DO 2000 L=1,L2
      DO 1995 LP=1,L2
 1995 BUFF1(KK1+LP) = ETABI(L,LP)
      KK1 = KK1 + L2
      DO 1996 LP=1,L2
 1996 BUFF1(KK1+LP) = ETAOI(L,LP)
      KK1 = KK1 + L2
      BUFF1(KK1+1) = DV(L)
      BUFF1(KK1+2) = XV(L)
      BUFF1(KK1+3) = YV(L)
      BUFF1(KK1+4) = CVIN(L)
```

```
      KK1 = KK1 + 4
 2000 CONTINUE
C
C
C     C O M P U T E   A N D   W R I T E   E T A B T
C     ETABT(L,LP)        = FRACTION OF (ENERGY IN FAN LP PASSING INTO CELL
C       (I,J) THRU  THE BOTTOM OF THE CELL) THAT PASSES OUT CELL (I,J)
C       THRU THE TOP IN FAN L      (L,LP MEASURED AT X(I))
 731    ISEND = 2
        LMDA = DY(J) * (SMU(M)/MU(M))
        XMIN = X(I-1)
        XMAX = X(I)
        AID = DY(J)*DY(J)*(SMU(M)/MU(M))*(SMU(M)/MU(M))
        HELP1 = AID - X(I-1)*X(I-1)
        HELP2 = AID - X(I)*X(I)
        DO 800 LP=1,L2
        EPS = TOLER * CH(LP) / (FORPI*W(M)*ABS(MU(M)))
        ANGL1 = X(I) * SINLP(LP)
        ANGL2 = X(I) * SINLP(LP-1)
C     ENTER INWARD , EXIT INWARD
        LIM1 = AMAX1( ANGL2, AMIN1( X(I),SQRT(LMDA*LMDA+X(I-1)*X(I-1))))
        IF ( LIM1 .LT. X(I) )  GO TO 733
        L = LP
        ETABT(L,LP) = 0.
        GO TO 800
 733    IAID = 1
        L = LP
        ETABT(L,LP) = SINT(LIM1,X(I),EPS,BOT) / CH(LP)
      . * FORPI * W(M) * ABS(MU(M))
 800    CONTINUE
C
        IF (DUGPNT .LT. 1.E-20)  GO TO 900
        WRITE (6,9004)  M,J,I
        DO 890 L=1,L2
        WRITE (6,9001)  L,(ETABT(L,LP),LP=1,L2)
 890    CONTINUE
 900    CONTINUE
C
C
C     C O M P U T E   A N D   W R I T E   E T A O T
C     ETAOT(L,LP)        = FRACTION OF (ENERGY IN FAN LP PASSING INTO CELL
C     (I,J) THRU THE OUTSIDE OF THE CELL) THAT PASSES OUT CELL (I,J) THRU
C     THE TOP IN FAN L
        ISEND = 2
        DELTY = DY(J)
        TANN = SMU(M)/MU(M)
        XMAX = X(I)
        XMIN = X(I-1)
        ASN = ASIN(X(I-1)/X(I))
        LIM1 = AMAX1( 0. , DY(J)-SQRT(X(I)*X(I)-X(I-1)*X(I-1))/TANN )
        DO 1200 LP=1,L2
        EPS = TOLER * CV(LP) / (FORPI*X(I)*W(M)*SMU(M))
        ANGL1 = PP * FLOAT(LP-1)
        ANGL2 = PP * FLOAT(LP)
        LIM2 = DY(J) - X(I)*COS(FLOAT(LP)*PP)/TANN
```

```
C   CASE1 -- EXIT INWARD
      IAID = 1
      L = LP
      ETAOT(L,LP) = SINT(LIM1,DY(J),EPS,OUT) / CV(LP)
     .   * FORPI * X(1) * W(M) * SMU(M)
 1200 CONTINUE
C
      IF (DUGPRT .LT. 1.E-20)  GO TO 1300
      WRITE (6,9005)  M,J,I
      DO 1290 L=1,L2
      WRITE (6,9001)  L,(ETAOT(L,LP),LP=1,L2)
 1290 CONTINUE
 1300 CONTINUE
C   COMPUTE  DH(L),XH(L),YH(L)
      LLOW = L2P1
      LHI = LMAX
      ISENDD = 2
      CALL  DISTNC
      DO 2500 L=1,L2
      DO 1997 LP=1,L2
 1997 BUFF1(KK1+LP) = ETABT(L,LP)
      KK1 = KK1 + L2
      DO 1998 LP=1,L2
 1998 BUFF1(KK1+LP) = ETAOT(L,LP)
      KK1 = KK1 + L2
      BUFF1(KK1+1) = DH(L)
      BUFF1(KK1+2) = XH(L)
      BUFF1(KK1+3) = YH(L)
      BUFF1(KK1+4) = CH(L)
      KK1 = KK1 + 4
 2500 CONTINUE
      IF (KK1 .GT. 1023)  STOP
      IF (KK2+KK1 .LT. 1023)  GO TO 2505
      WRITE (3)  (BUFF2(ILM),ILM=1,1023)
      NOBUFF = NOBUFF + 1
      KK2 = 1
      NOO = 0
 2505 DO 2510 KKK=1,KK1
 2510 BUFF2(KK2+KKK) = BUFF1(KKK)
      KK2 = KK2 + KKI
      NOO = NOO + 1
 2600 CONTINUE
      DO 2002 LP=1,L2
      DO 2002 L=1,L2
      SUMB(I,LP) = SUMB(I,LP) + ETABI(L,LP) + ETABT(L,LP)
 2002 SUMO(I,LP) = SUMO(I,LP) + ETAOI(L,LP) + ETAOT(L,LP)
      RETURN
C
C
 9000 FORMAT (1H0/20H ETABI(L,LP)  FOR M=,I2,3H J=,I2,3H I=,I2/60X,2HLP)
 9001 FORMAT (3H L=,I2/(6X,1P6E14.6))
 9003 FORMAT (1H0/20H ETAOI(L,LP)  FOR M=,I2,3H J=,I2,3H I=,I2/60X,2HLP)
 9004 FORMAT (1H0/20H ETABT(L,LP)  FOR M=,I2,3H J=,I2,3H I=,I2/60X,2HLP)
 9005 FORMAT (1H0/20H ETAOT(L,LP)  FOR M=,I2,3H J=,I2,3H I=,I2/60X,2HLP)
      END
```

Subroutine OUTWRD

```
      SUBROUTINE   OUTWRD
C     GIVEN INDICES  M,J,I    COMPUTE, PRINT, AND SAVE
C         1)  ETAIO(L,LP)                 LP.GE.L.GT.L2
C         2)  ETABO(L,LP)                 L.GT.L2 , LP=1,LMAX
C         3)  ETAOO(L,LP)                 L.GT.L2 , LP=1,L2
C         4)  ETAIT(L,LP)                 LP.GE.L.GT.L2
C         5)  ETABT(L,LP)                 L.GT.L2, LP=1,LMAX (EXITING
C                                                             OUTWARD)
C         6)  ETAOT(L,LP)                 L.GT.L2 , LP=1,LMAX (EXITING
C                                                             OUTWARD)
C         7)  DV(L),XV(L),CVIN(L),CVOUT(L),DH(L),XH(L),CH(L)
C
C     ****************************************************************DHA8007
C          H E C T I C   C O M M O N                                  DHA8008
C     ****************************************************************DHA8009
      COMMON   AMN,     AMBIEN, ANN,    ATOM,   BCATAG, BCBTAG, BCLTAG,  HECCOM01
     1BCHTAG,  CAPIN,   CAPS,   CDUT,   CMXK,   CMXOM,  CNDE,   CO,      HECCOM02
     2COL,     CSTOP,   CCVV,   CYCLE,  DBGPRT, DGHET,  DHNU,   DIANTP,  HECCOM03
     3DMIN,    DNN,     DT,     DTDUGR, DTC,    DTH,    DTNA,   DTO,     HECCOM04
     4DTH,     DTR1,    DTR2,   DTUF,   DTVF,   DUMPT7, ECK,    EFRAC,   HECCOM05
     5E11,     ERRCRT,  ETH,    EZERO,  FDTC,   FFA,    FFB,    FIFT,    HECCOM06
     6FUDGE,   GV,      HCB,    HCP,    HH,     HNU,    HNUP,   HNUS,    HECCOM07
     7HVB,     I,       IC,     IGOTO,  IH,     IHNU,   IMAX,   IMPTAG,  HECCOM08
     8ISSSS,   ISR,     ITAG,   ITHMAX, IU,     IV,     I1,     I2,      HECCOM09
     9J,       JC,      JH,     JMAX,   JMAXA,  JU,     JV,     J5       HECCOM10
      COMMON   K,       KMAXA,  KMN1,   KMN2,   KP,     LLLL,   MMMM,    HECCOM11
     1MERGE,   MFTAG,   MZ,     N,      NC,     NHNU,   NK,     NPC,     HECCOM12
     2NN,      NRM,     NT,     NTAPE,  NY,     N1,     N2,     N3,      HECCOM13
     3NN,      ODDC,    PABOVE, PBLO,   PI,     PIDTS,  PRINTL, PRINTS,  HECCOM14
     4PHOR,    PRR,     PW,     RADE,   RC,     RFT,    RPTAG,  RR,      HECCOM15
     5SCUR,    SCR,     SCHE,   SCYCLE, SGNL,   SIG,    SIGH,   SLUG,    HECCOM16
     6SN,      SPROB,   SVMAX,  SVS,    S1,     S2,     S3,     S4,      HECCOM17
     7SS,      T,       TAUDTS, THICK,  TMAX,   TRAD,   URR,    UT,      HECCOM18
     8UU,      UVMAX,   VABOVE, VAPE,   VBLO,   VEL,    WS,     WSA,     HECCOM19
     9XMAXX,            BLANK(49)                                        HECCOM20
      COMMON            A1X(1200),      AMX(1200),      ALAMH(1200),
     1ALAMV(1200),      B(1200),        CAP(1200),      ER(1200),
     2FIOUT(1200),      KFIT(1200),     P(1200),        SMLQ(1200),
     3THETA(1200),      U(1200),        V(1200),        XALP(1200),
     4XB(1200),         SOLID(400),     PL(200),        Y(101),
     5DT(100),          GAMC(100),      YAMC(100),      UL(100),         HECCOM26
     6W2(100),          X(53),          DX(52),         TAU(52),         HECCOM27
     7HETAB(50),        TEMP(12),       HEAD(12)                         HECCOM28
C                                                                        DRA8030
C     THE FOLLOWING VARIABLES IN HECTIC COMMON HAVE BEEN RENAMED IN THIS
C     ROUTINE  --  CV TO CCVV , ISEND TO ISSSS , L TO LLLL , M TO MMMM,
C     AND XMAX TO XMAXX
C     ****************************************************************DRA8039
C                                                                     DHA8040
      DIMENSION       FLEFT(100),     PR(100),        SIGC(100)       DRA8041
      EQUIVALENCE     (FLEFT,UL),     (PL,PR)
      EQUIVALENCE     (PL(101),SIGC)
      EQUIVALENCE (S20,BLANK(12)), (ZP120,BLANK(13)), (ZP136,BLANK(14)),
     1(KMX,BLANK(15)),(NT,BLANK(16)),(ANUMBR,BLANK(19)),(LOCZ,BLANK(22))
     2,(M,BLANK(25)),(NA,BLANK(26))
      EQUIVALENCE (BACC,BLANK(47)),(SACC,BLANK(48)),(TACC,BLANK(49))
```

```
C                                                                       DHA8045
      REAL  MERGE                                                       DHA8046
C                                                                       DHA8047
C                                                                       DHA8048
      COMMON    /RAD1/    IAIU,     KMAX,     ICHECK,   ISEND,          RAD1
     1L2,       L2P1,     LMAX,     M2,       MUMAX,    ISENDD,         RAD2
     2ISWEEP,   LLOW,     LMI                                           RAD
      COMMON    /RAD2/    ANGL1,    ANGL2,    ASN,      CHKSUM,         RAD3
     1DELTY,    FOURPI,   HELP1,    HELP2,    LMDA,     PP,             RAD4
     2PI2,      PI4,      SINP2,    TANN,     TOLER,    XMIN,           RAD5
     3XMAX,     STEF                                                    RAD6
      COMMON    /RAD3/              DUM1(1),            AA(6),          RAD7
     1          COSPHI(6),          CVIN(6),            CVOUT(6),       RAD8
     2          CH(6),              MU(6),              PHI(6),         RAD9
     3          DUM2(1),            SINLP(6),           SMU(6),         RAD10
     4          W(6),               CV(6)                               RAD11
      COMMON   /INDEX/   M,L,LP                                         RAD12
      COMMON    /DRUMM/   LCHECK,   KK2,      NOBUFF                    RAD13
      COMMON    /DIS/     DV(6),    XV(6),    TV(6),    DH(6),          RAD14
     1          XH(6),    YH(6)                                         RAD15
      COMMON    /SUMCHK/            SUMB(50,6),         SUMO(50,6)      RAD16
      COMMON    /BNDT/    BTHETA(50),     RTHETA(50),     ATHETA(50)    RAD17
C
      REAL    MU , LMDA
C
      EQUIVALENCE    (FIOUT,OLDTH) , (U,ROSS) ,
     .   (CAP(1),BUFF1(1),BUFF(1),NOI) , (P(1),PLANCK(1),BUFF2(1),NOO)
      DIMENSION   OLDTH(1) , ROSS(1) , BUFF1(1) , BUFF(1) , PLANCK(1) ,
     .  BUFF2(1)
C
      EQUIVALENCE    (V(1),XA(1),EB(1))
      DIMENSION   XA(1)
      DIMENSION              EB(50,6) , EV(50,6) , ET(50,6) ,
     .           ETAIO(6,6) , ETAIT(6,6) , ETABI(6,6) , ETABT(6,6),
     .  ETABO(6,6) , ETAOI(6,6) , ETAOT(6,6) , ETAOO(6,6)
C
C
C
      REAL  JO , JB
      REAL   LIM1 , LIM2
      EXTERNAL   BOT
      EXTERNAL   OUT
      EXTERNAL   IN
      REAL   IN
C
C   C O M P U T E   S P E C I F I E D   C O N S T A N T S
C
      DO 40 L=0,LMAX
   40 AA(L) = SINLP(L) * X(I)
      DO 50 LP=L2P1,LMAX
   50 CVIN(LP) = JO(J,I-1,LP,M)
      DO 60 LP=1,LMAX
   60 CVOUT(LP) = JO(J,1,LP,M)
      DO 70 LP=1,LMAX
   70 CH(LP) = JH(J-1,1,LP,M)
```

```
      ISWELP = 2
C
C   Z E R O   E T A S
      DO 500 L=L2PI,LMAX
      DO 500 LP=1,LMAX
      ETAIO(L,LP) = 0.
      ETAIT(L,LP) = 0.
      ETABO(L,LP) = 0.
      ETAOO(L,LP) = 0.
      ETABT(L,LP) = 0.
      ETAOT(L,LP) = 0.
  500 CONTINUE
C
C
C   C O M P U T E   A N D   W R I T E   E T A I O , E T A I T
C   ETAIO(L,LP)       = FRACTION OF (ENERGY IN FAN LP PASSING INTO CELL
C   (I,J) THRU THE INSIDE OF THE CELL) THAT PASSES OUT CELL (I,J) THRU
C   THE OUTSIDE IN FAN L
C   ETAIT(L,LP)        = FRACTION FROM INSIDE CELL (I,J) TO TOP
C   SUM OF ABOVE TWO FRACTIONS = 1.
      IF (I .EQ. 1)  GO TO 2080
      DELTY = DY(J)
      TANN = SMU(M)/MU(M)
      XMAX = X(I)
      XMIN = X(I-1)
      HELP1 = X(I-1)*X(I-1) - X(I)*X(I)
      LP = LMAX
      EPS = TOLER * CVIN(LP) / (FORPI*X(I-1)*W(M)*SMU(M))
      DQ = 4. * EPS
      L = LMAX
      R = 0.
      Q = 0.
 2020 QP = Q
      RP = R
      IF ( X(I)*SINLP(L-1)  .LT.  X(I-1)*SINLP(LP-1)  )  GO TO 2030
C   LP DOES NOT FILL L
      XI = SINLP(LP-1)
      ANGL1 = PP * FLOAT(LP-1)
      LD = 0
      GO TO 2040
C   LP AT LEAST FILLS L
 2030 XI = X(I) * SINLP(L-1)/X(I-1)
      ANGL1 = PI - ASIN(XI)
      LD = 1
 2040 R = DY(J) * XI
      IF ( DQ.LT.(2.*EPS)  .OR.  DX(I)+(2.*EPS).GT.DY(J)*TANN )
     .  GO TO 2050
C   OUTSIDE WALL RECEIVES SOME RADIATION FROM INSIDE
      Q = SINT(0.,DY(J),EPS,IN)
      DQ = Q - QP
      ETAIO(L,LP) = DQ / CVIN(LP)
     .  * FORPI * X(I-1) * W(M) * SMU(M)
      GO TO 2060
 2050 DQ = 0.
 2060 ETAIT(L,LP) = (R - RP - DQ) / CVIN(LP)
     .  * FORPI * X(I-1) * W(M) * SMU(M)
```

```
      LP = LP - 1 + LD
      IF ( LP .EQ. L2 )  GO TO 2100
      EPS = TOLER * CVIN(LP) / (FORPI*X(I-1)*W(M)*SMU(M))
      L = L - LD
      GO TO 2020
 2080 DO 2090 L=L2P1,LMAX
      DO 2090 LP=L2P1,LMAX
      ETAIO(L,LP) = 0.
 2090 ETAIT(L,LP) = 0.
 2100 CONTINUE
C
      IF (DUGPRT .LT. 1.E-20)  GO TO 2150
      WRITE (6,8003)  M,J,I
      DO 2110 L=L2P1,LMAX
      WRITE (6,8001)  L,(ETAIO(L,LP),LP=1,LMAX)
 2110 CONTINUE
      WRITE (6,8004)  M,J,I
      DO 2120 L=L2P1,LMAX
      WRITE (6,8001)  L,(ETAIT(L,LP),LP=1,LMAX)
 2120 CONTINUE
 2150 CONTINUE
C
C   C H E C K S U M   E T A I O ,  E T A I T
C
      IF (I .EQ. 1)  GO TO 2210
      DO 2200 LP=L2P1,LMAX
      SUMOO = 0.
      SUMTT = 0.
      SUM = 0.
      DO 2198 L=L2P1,LMAX
      SUMOO = SUMOO + ETAIO(L,LP)
      SUMTT = SUMTT + ETAIT(L,LP)
 2198 CONTINUE
      SUM = SUMOO + SUMTT
      IF (ABS(SUM-1.0)  .LT.  CHKSUM)  GO TO 2200
      ICHECK = 1
      WRITE (6,9017)  M,J,I
      WRITE (6,9018)  LP,SUMTT,SUMOO SUM
 2200 CONTINUE
C
C
C
C   C O M P U T E   A N D   W R I T E   E T A B O
C   ETABO(L,LP)        = FRACTION OF (ENERGY IN FAN LP PASSING INTO CELL
C   (I,J) THRU THE BOTTOM OF THE CELL) THAT PASSES OUT CELL (I,J)
C   THRU THE OUTSIDE IN FAN L     (L,LP MEASURED AT X(I))
 2210 ISEND = 2
      LMDA = DY(J) * (SMU(M)/MU(M))
      XMIN = X(I-1)
      XMAX = X(I)
      AID = DY(J)*DY(J)*(SMU(M)/MU(M))*(SMU(M)/MU(M))
      HELP1 = AID - X(I-1)*X(I-1)
      HELP2 = AID - X(I)*X(I)
      DO 950 LP=1,LMAX
      EPS = TOLER * CH(LP) / (FORPI*W(M)*ABS(MU(M)))
```

```
      ANGL1 = X(I) * SINLP(LP)
      ANGL2 = X(I) * SINLP(LP-1)
      IF (LP .GT. L2)  GO TO 925
C   ENTER INWARD , EXIT OUTWARD
      LIM1 = AMAX1( ANGL2 , X(I-1) )
      IAID = 4
      L = LMAX + I - LP
      ETABO(L,LP) = SINT(LIM1,X(I),EPS,BOT) / CH(LP)
     .  * FORPI * W(M) * ABS(MU(M))
      GO TO 949
C   ENTER OUTWARD , EXIT OUTWARD
 925  LIM1 = AMAX1( ANGL1 , X(I-1) )
      IAID = 5
      L = LP
      ETABO(L,LP) = SINT(LIM1,X(I),EPS,BOT) / CH(LP)
     .  * FORPI * W(M) * ABS(MU(M))
 949  CONTINUE
 950  CONTINUE
C
      IF (DUGPRT .LT. 1.E-20)  GO TO 970
      WRITE (6,8005)  M,J,I
      DO 960 L=L2P1,LMAX
      WRITE (6,8001)  L,(ETABO(L,LP),LP=1,LMAX)
 960  CONTINUE
 970  CONTINUE
C
C
C   C O M P U T E   A N D   W R I T E   E T A O O
C   ETAOO(L,LP)        = FRACTION OF (ENERGY IN FAN LP PASSING INTO CELL
C   (I,J) THRU THE OUTSIDE OF THE CELL)  THAT PASSES OUT CELL (I,J) THRU
C   THE OUTSIDE IN FAN L
      ISEND = 3
      DELTY = DY(J)
      TANN = SMU(M)/MU(M)
      XMAX = X(I)
      XMIN = X(I-1)
      ASN = ASIN(X(I-1)/X(I))
      DO 1300 LP=1,L2
      ANGL1 = PP * FLOAT(LP-1)
      ANGL2 = PP * FLOAT(LP)
      L = LMAX + 1 - LP
      LIM2 = DY(J) - 2.*X(I)*COS(PP*FLOAT(LP))/TANN
      IF (LIM2 .LT. 0.)  GO TO 1230
      EPS = TOLER * CVOUT(LP) / (FORPI*X(I)*W(M)*SMU(M))
      ETAOO(L,LP) = SINT(0.,LIM2,EPS,OUT) / CVOUT(LP)
     .  * FORPI * X(I) * W(M) * SMU(M)
      GO TO 1240
 1230 ETAOO(L,LP) = 0.
 1240 CONTINUE
 1300 CONTINUE
C
      IF (DUGPRT .LT. 1.E-20)  GO TO 1320
      WRITE (6,8006)  M,J,I
      DO 1310 L=L2P1,LMAX
      WRITE (6,8001)  L,(ETAOO(L,LP),LP=1,L2)
```

```
 1310 CONTINUE
 1320 CONTINUE
C     COMPUTE  DV(L),XV(L),YV(L)
      LLOW = 1
      LHI = L2
      ISENDD = 1
      CALL  DISTNC
      KK1 = 0
      DO 3000 L=L2P1,LMAX
      DO 2990 LP=L2P1,LMAX
 2990 BUFF1(KK1+LP-L2) = ETAIO(L,LP)
      KK1 = KK1 + L2
      DO 2991 LP=1,LMAX
 2991 BUFF1(KK1+LP) = ETABO(L,LP)
      KK1 = KK1 + LMAX
      DO 2992 LP=1,L2
 2992 BUFF1(KK1+LP) = ETAOO(L,LP)
      KK1 = KK1 + L2
      BUFF1(KK1+1) = DV(L)
      BUFF1(KK1+2) = XV(L)
      BUFF1(KK1+3) = YV(L)
      BUFF1(KK1+4) = CVOUT(L)
      KK1 = KK1 + 4
 3000 CONTINUE
C
C
C     C O M P U T E    A N D    W R I T E    E T A B T    (EXITING OUTWARD)
C     ETABT(L,LP)        = FRACTION OF (ENERGY IN FAN LP PASSING INTO CELL
C       (I,J) THRU  THE BOTTOM OF THE CELL) THAT PASSES OUT CELL (I,J)
C       THRU THE TOP IN FAN L      (L,LP MEASURED AT X(I))
      ISEND = 2
      LMDA = DY(J) * (SMU(M)/MU(M))
      XMIN = X(I-1)
      XMAX = X(I)
      AID = DY(J)*DY(J)*(SMU(M)/MU(M))*(SMU(M)/MU(M))
      HELP1 = AID - X(I-1)*X(I-1)
      HELP2 = AID - X(I)*X(I)
      DO 800 LP=1,LMAX
      EPS = TOLER * CH(LP) / (FORPI*W(M)*ABS(MU(M)))
      ANGL1 = X(I) * SINLP(LP)
      ANGL2 = X(I) * SINLP(LP-1)
      IF (LP .GT. L2)  GO TO 740
C     ENTER INWARD , EXIT OUTWARD
  735 LIM1 = AMAX1( X(I-1) , ANGL2 )
      IAID = 2
      L = LMAX + 1 - LP
      ETABT(L,LP) = SINT(LIM1,X(I),EPS,BOT) / CH(LP)
     .  * FORPI * W(M) * ABS(MU(M))
      GO TO 800
  740 CONTINUE
C     ENTER OUTWARD , EXIT OUTWARD
      LIM1 = AMAX1( X(I-1) , ANGL1 )
      IAID = 3
      L = LP
      ETABT(L,LP) = SINT(LIM1,X(I),EPS,BOT) / CH(LP)
```

```
     .  * FORPI * W(M) * ABS(MU(M))
 800  CONTINUE
C
      IF (DBGPRT .LT. 1.E-20)  GO TO 900
      WRITE (6,8000)  M,J,I
      DO 830 L=L2P1,LMAX
      WRITE (6,8001)  L,(ETABT(L,LP),LP=1,LMAX)
 830  CONTINUE
 900  CONTINUE
C
C
C   C O M P U T E   A N D   W R I T E   E T A O T   (EXITING OUTWARD)
C   ETAOT(L,LP)        = FRACTION OF (ENERGY IN FAN LP PASSING INTO CELL
C   (I,J) THRU THE OUTSIDE OF THE CELL) THAT PASSES OUT CELL (I,J) THRU
C   THE TOP IN FAN L
C   CASE2 -- EXIT OUTWARD
      ISEND = 2
      IAID = 2
      DELTY = DY(J)
      TANN = SMU(M)/MU(M)
      XMAX = X(I)
      XMIN = X(I-1)
      ASN = ASIN(X(I-1)/X(I))
      DO 1200 LP=1,L2
      L = LMAX + 1 - LP
      LIM2 = DY(J) - X(I)*COS(FLOAT(LP)*PP)/TANN
      IF (LIM2 .LT. 0.)  GO TO 1130
      ANGL1 = PP * FLOAT(LP-1)
      ANGL2 = PP * FLOAT(LP)
      EPS = TOLER * CVOUT(LP) / (FORPI*X(I)*W(M)*SMU(M))
      LIM1 = AMAX1( 0. , DELTY-2.*X(I)*COS(FLOAT(LP-1)*PP)/TANN)
      ETAOT(L,LP) = SINT(LIM1,LIM2,EPS,OUT) / CVOUT(LP)
     .  * FORPI * X(I) * W(M) * SMU(M)
      GO TO 1200
 1130 ETAOT(L,LP) = 0.
 1200 CONTINUE
C
      IF (DBGPRT .LT. 1.E-20)  GO TO 1220
      WRITE (6,8002)  M,J,I
      DO 1210 L=L2P1,LMAX
      WRITE (6,8001)  L,(ETAOT(L,LP),LP=1,L2)
 1210 CONTINUE
 1220 CONTINUE
C   COMPUTE  DH(L),XH(L),YH(L)
      LLOW = 1
      LHI = L2
      ISENDD = 2
      CALL  DISTNC
      DO 3500 L=L2P1,LMAX
      DO 3001 LP=L2P1,LMAX
 3001 BUFF1(KK1+LP-L2) = ETAIT(L,LP)
      KK1 = KK1 + L2
      DO 3005 LP=1,LMAX
 3005 BUFF1(KK1+LP) = ETABT(L,LP)
      KK1 = KK1 + LMAX
```

```
      DO 3003 LP=1,L2
 3003 BUFF1(KK1+LP) = ETAOT(L,LP)
      KK1 = KK1 + L2
      BUFF1(KK1+1) = DH(L)
      BUFF1(KK1+2) = XH(L)
      BUFF1(KK1+3) = TH(L)
      BUFF1(KK1+4) = CH(L)
      KK1 = KK1 + 4
 3500 CONTINUE
      IF (KK1 .GT. 1023)  STOP
      IF (KK2+KK1 .LT. 1023)  GO TO 3505
      WRITE (3)  (BUFF2(ILM),ILM=1,1023)
      NOBUFF = NOBUFF + 1
      KK2 = 1
      NOO = 0
 3505 DO 3510 KKK=1,KK1
 3510 BUFF2(KK2+KKK) = BUFF1(KKK)
      KK2 = KK2 + KK1
      NOO = NOO + 1
 3600 CONTINUE
      DO 3100 LP=1,LMAX
      DO 3100 L=L2P1,LMAX
      SUMB(1,LP) = SUMB(1,LP) + ETABO(L,LP) + ETABT(L,LP)
 3100 SUMO(1,LP) = SUMO(1,LP) + ETAOO(L,LP) + ETAOT(L,LP)
      RETURN
C
 8000 FORMAT (1H0/40H ETABT(L,LP)  (SWEEPING OUTWARD)  FOR M=,I2,3H J=,I
     .2,3H I=,I2/60X,2HLP)
 8001 FORMAT (3H L=,I2/(6X,1P8E14.6))
 8002 FORMAT (1H0/40H ETAOT(L,LP)  (SWEEPING OUTWARD)  FOR M=,I2,3H J=,I
     .2,3H I=,I2/60X,2HLP)
 8003 FORMAT (1H0/20H ETAIO(L,LP)  FOR M=,I2,3H J=,I2,3H I=,I2/60X,2HLP)
 8004 FORMAT (1H0/20H ETAIT(L,LP)  FOR M=,I2,3H J=,I2,3H I=,I2/60X,2HLP)
 8005 FORMAT (1H0/20H ETABO(L,LP)  FOR M=,I2,3H J=,I2,3H I=,I2/60X,2HLP)
 8006 FORMAT (1H0/20H ETAOO(L,LP)  FOR M=,I2,3H J=,I2,3H I=,I2/60X,2HLP)
 9017 FORMAT (26H0SUM OF INSIDE ETAS FOR M=,I2,4H, J=,I2,4H, I=,I2,12H,
     .OVER L FOR/15X,4HSUMT,11X,4HSUMO,11X,3HSUM/)
 9018 FORMAT (4X,3HLP=,I2,1P3E15.6)
 9019 FORMAT (29H0CHECKSUMS FOR INSIDE ETAS=1.)
      END
```

Subroutine DISTNC

```
      SUBROUTINE   DISTNC
C     GIVEN INDICES  M,J,I  COMPUTE  DV(L),XV(L),DH(L),XH(L)
C     NOTE -- M  MUST BE .LE. M2
C      ***************************************************************DRA8007
C         H E C T I C   C O M M O N                                   DRA8008
C      ***************************************************************DRA8009
      COMMON  AHN,    AMBIEN, ANH,    ATOM,   BCATAG, BCBTAG, BCLTAG,  HECCOM01
     1BCRTAG, CAPIN,  CAPS,   CDUT,   CMXK,   CMXOM,  CNDE,   CO,      HECCOM02
     2COE,    CSTOP,  CCVV,   CYCLE,  OBGPRT, DGREY,  DHMU,   OIANTP,  HECCOM03
     3DMIN,   DNN,    DT,     DTBUGR, OTC,    OTH,    OTNA,   OTQ,     HECCOM04
     4DT?,    DTH1,   DTH2,   DTUF,   DTVF,   DUMPT7, ECK,    EFRAC,   HECCOM05
     5E11,    ERRCHT, ETH,    EZENO,  FOIC,   FFA,    FFB,    FIFT,    HECCOM06
     6FUDGE,  GV,     HCH,    HCP,    HH,     HNU,    HNUP,   HNUS,    HECCOM07
     7HVB,    II,     IC,     IGOTO,  IH,     IHNU,   IMAX,   IMPTAG,  HECCOM08
     8ISSSS,  ISR,    ITAG,   ITRMAX, IU,     IV,     I1,     I2,      HECCOM09
     9J,      JC,     JH,     JMAX,   JMAXA,  JU,     JV,     J5       HECCOM10
      COMMON  K,      KMAXA,  KMN1,   KMN2,   KP,     LLLL,   MMMM,    HECCOM11
     1MERGE,  MFTAG,  MZ,     N,      NC,     NMNU,   NK,     NPC,     HECCOM12
     2NR,     NNM,    NT,     NTAPE,  NY,     N1,     N2,     N3,      HECCOM13
     3NN,     OUDC,   PABOVE, PBLO,   PI,     P1OTS,  PRINTL, PRINTS,  HECCOM14
     4PROB,   PRR,    PW,     RADE,   RC,     RFT,    RPTAG,  RR,      HECCOM15
     5SCDR,   SCR,    SCHE,   SCYCLE, SGNL,   SIG,    SIGH,   SLUB,    HECCOM16
     6SH,     SPROB,  SVMAX,  SVS,    S1,     S2,     S3,     S4,      HECCOM17
     7SS,     T,      TAUUTS, THICK,  TMAX,   TRAD,   URH,    UT,      HECCOM18
     8UU,     UVMAX,  VABOVE, VAPE,   VBLO,   VEL,    WS,     WSA,     HECCOM19
     9XMAXX,          BLANK(49)                                        HECCOM20
      COMMON          AIX(1200),      AMX(1200),      ALAMH(1200),
     1ALAMV(1200),    B(1200),        CAP(1200),      ER(1200),
     2FIOUT(1200),    KFIT(1200),     P(1200),        SMLQ(1200),
     3THETA(1200),    U(1200),        V(1200),        XALP(1200),
     4XB(1200),       SOLIO(400),     PL(200),        Y(101),
     5DY(100),        GAMC(100),      YAMC(100),      UL(100),         HECCOM26
     6W2(100),        X(53),          DX(52),         TAU(52),         HECCOM27
     7BETAB(50),      TEMP(12),       HEAD(12)                         HECCOM28
C                                                                      DRA8038
C     THE FOLLOWING VARIABLES IN HECTIC COMMON HAVE BEEN RENAMED IN THIS
C     ROUTINE  --  CV TO CCVV , ISEND TO ISSSS , L TO LLLL , M TO MMMM,
C     XMAX TO XMAXX , AND I TO II
C      ***************************************************************DRA8039
C                                                                      DRA8040
      DIMENSION       FLEFT(100),     PR(100),        SIGC(100)        DRA8041
      EQUIVALENCE     (FLEFT,UL),     (PL,PR)
      EQUIVALENCE    (PL(101),SIGC)
      EQUIVALENCE (S20,BLANK(12)), (ZP126,BLANK(13)), (ZP136,BLANK(14)),
     1(KMX,BLANK(15)),(WT,BLANK(16)),(ANUMBR,BLANK(19)),(LOCZ,BLANK(22))
     2,(H,BLANK(25)),(NA,BLANK(26))
      EQUIVALENCE (BACC,BLANK(47)),(SACC,BLANK(48)),(TACC,BLANK(49))
C                                                                      DRA8045
      REAL  MERGE                                                      DRA8046
C                                                                      DRA8047
C                                                                      DRA8048
      COMMON     /RAD1/     IAIO,      KMAX,      ICHECK,    ISEND,    RAD1
     1L2,        L2P1,      LMAX,      M2,        MUMAX,     ISENDD,   RAD2
     2ISWEEP,    LIM1,      LIM2                                       RAD
      COMMON     /RAD2/     ANGL1,     ANGL2,     ASN,       CHKSUM,   RAD3
     1DELTY,     FORP1,     HELP1,     HELP2,     LMDA,      PP,       RAD4
```

```
     2PI2,      PI4,       SINP2,     TANN,      TOLER,     XMIN,            RAD5
     3XMAX,     STEF                                                        RAD6
      COMMON    /RAD3/                DUM1(1),              AA(16),          RAD7
     1          COSPHI(6),            CVIN(6),              CVOUT(6),        RAD8
     2          CH(6),                MU(6),                PHI(6),          RAD9
     3          DUM2(1),              SINLP(6),             SMU(6),          RAD10
     4          W(6),                 CV(6)                                  RAD11
      COMMON   /INDEX/   M,L,LP                                             RAD12
      COMMON    /DRUMM/   LCHECK,    KK2,       NOBUFF                       RAD13
      COMMON    /DIS/     DV(6),     XV(6),     YV(6),      DH(6),           RAD14
     1          XH(6),    YH(6)                                              RAD15
      COMMON    /SUMCHK/              SUMB(50,6),           SUMO(50,6)       RAD16
      COMMON    /BNDT/    BTHETA(50),      RTHETA(50),    ATHETA(50)         RAD17
C
      REAL    MU , LMDA
C
      EQUIVALENCE    (FIOUT,OLDTH) , (U,ROSS) ,
     .    (CAP(1),BUFF1(1),BUFF(1),NOI) , (P(1),PLANCK(1),BUFF2(1),NOO)
      DIMENSION   OLDTH(1) , ROSS(1) , BUFF1(1) , BUFF(1) , PLANCK(1) ,
     .  BUFF2(1)
C
      EQUIVALENCE    (V(1),XA(1),EB(1))
      DIMENSION   XA(1)
      DIMENSION             EB(50,6) , EV(50,6) , ET(50,6) ,
     .          ETAIO(6,6) , ETAIT(6,6) , ETABI(6,6) , ETABT(6,6),
     .  ETABO(6,6) , ETAOI(6,6) , ETAOT(6,6) , ETAOO(6,6)
C
C
C
C    NOTE -- IN COMMON, THE FOLLOWING VARIABLES HAVE CHANGED NAMES IN
C    THIS ROUTINE
C      LLOW    GOES TO    LIM1
C      LHI    GOES TO    LIM2
C
      I = I1
      GO TO (5,550), ISENDD
C
C    DV(L)=                  L.LE.L2(SWEEPING INWARD)
C         CHARACTERISTIC LENGTH OF FAN (L,M) IN CELL (I,J) MEASURED FROM
C         X(I-1),Y(J-.5)  IN DIRECTION  PHI(LMAX+1-L),(PI-ACOS(MU(M)))
C         TO THE NEAREST CELL FACE
C    DV(L)=                  L.GT.L2(SWEEPING OUTWARD)
C         CHARACTERISITC LENGTH OF FAN (L,M) IN CELL (I,J) MEASURED FROM
C         X(I),Y(J-.5)  IN DIRECTION  PHI(LMAX+1-L),(PI-ACOS(MU(M)))  TO
C         THE NEAREST CELL FACE
C    XV(L)=                  ALL L
C         THE X-COORDINATE OF THE INTERSECTION OF DV(L) AND THE CELL
C         FACE
C    YV(L)=                  ALL L
C         Y(J-.5) - (THE Y-COORDINATE OF THE INTERSECTION OF DV(L) AND
C         THE CELL FACE)
C
 5    IF (ISWEEP .EQ. 1)  I=I-1
      IF (DBGPRT .LT. 1.E-20)  GO TO 10
      WRITE (6,8000)  M,II,J
```

```
10   DO 500 L=LIM1,LIM2
C    CONSIDER LENGTHS IN T-PLANE
      IBACK = 0
      IF (L .GT. L2)  GO TO 250
      IF (I .LE. 1)  GO TO 50
      IF ( PHI(L)  .GT.  ASIN( X(I-1)/X(I) ) )  GO TO 50
C    HITS X(I-1) CIRCLE
      HIT = X(I-1)
      BBB = -1.
      GO TO 340
C    HITS X(I) CIRCLE
 50   HIT = X(I)
      IBACK = 1
      DVV = X(I) * COSPHI(L) * 2.
      GO TO 355
 250  IF (I .EQ. IMAX)  GO TO 500
C    HITS X(I+1) CIRCLE
      HIT = X(I+1)
      BBB = +1.
 340  A = X(I) * COSPHI(L)
      B = SQRT ( X(I)*X(I)*(COSPHI(L)*COSPHI(L)-1.) + HIT*HIT )
      DVV = A + B * BBB
 355  IF (DBGPRT .LT. 1.E-20)  GO TO 360
      K = LMAX +1 - L
      WRITE (6,8001)  K,DVV
C    CONSIDER 3-D LENGTHS
 360  IF ( DVV/(SMU(M)/MU(M))  .GT.  DT(J)/2. )  GO TO 370
C    HITS SIDE FIRST
      DV(LMAX+1-L) = DVV/SMU(M)
      XV(LMAX+1-L) = HIT
      TV(LMAX+1-L) = DVV/(SMU(M)/MU(M))
      GO TO 500
C    HITS BOTTOM FIRST
 370  DV(LMAX+1-L) = DT(J)/(2.*MU(M))
      HELP = DT(J)/2. * (SMU(M)/MU(M))
      IF ( IBACK.EQ.1  .AND.  HELP.GT.DVV/2. )  HELP=DVV-HELP
      XV(LMAX+1-L) = SQRT ( X(I)*X(I) + HELP*HELP - 2.*X(I)*HELP*
     .    COSPHI(L) )
      TV(LMAX+1-L) = DT(J)/2.
 500  CONTINUE
C
      IF (DBGPRT .LT. 1.E-20)  GO TO 1000
      LIM1P = LMAX +1 -LIM2
      LIM2P = LMAX +1 -LIM1
      WRITE (6,8002)
      WRITE (6,8001)  (L,DV(L),L=LIM1P,LIM2P)
      WRITE (6,8003)
      WRITE (6,8001)  (L,XV(L),L=LIM1P,LIM2P)
      WRITE (6,8006)
      WRITE (6,8001)  (L,TV(L),L=LIM1P,LIM2P)
 548  GO TO 1000
C
C    DH(L)=
C         CHARACTERISTIC LENGTH IN FAN (L,M) IN CELL (I,J) MEASURED FROM
C         X(I-.5),T(J)  IN DIRECTION  PHI(LMAX+1-L),(PI-ACOS(MU(M)))  TO
```

```
C          THE NEAREST CELL FACE
C     XH(L)=                      ALL L
C          THE X-COORDINATE OF THE INTERSECTION OF DH(L) AND THE CELL
C          FACE
C     YH(L)=                      ALL L
C          Y(J-.5) - (THE Y COORDINATE OF THE INTERSECTION OF DH(L) AND
C          THE CELL FACE)
C
  550  IF (DBGPRT .LT. 1,E-20)  GO TO 555
       WRITE (6,8004)
  555  A = X(I-1) + DX(I)/2.
       DO 590 L=LIM1,LIM2
C     CONSIDER LENGTHS IN Y-PLANE
       IBACK = 0
       IF (L .GT. L2)  GO TO 560
       IBACK = 1
       IF (I .EQ. 1)  GO TO 560
       IF ( PHI(L)  .GT.  ASIN( X(I-1)/A ) )  GO TO 560
C     HITS X(I-1) CIRCLE
       HIT = X(I-1)
       BBB = -1.
       GO TO 565
C     HITS X(I) CIRCLE
  560  HIT = X(I)
       BBB = +1.
  565  DHH = A*COSPHI(L) + BBB * SQRT( A*A*(COSPHI(L)*COSPHI(L)-1. ) +
     .      HIT*HIT )
       IF (DBGPRT .LT. 1.E-20)  GO TO 567
       K = LMAX +1 - L
       WRITE (6,8001)  K,DHH
C     CONSIDER 3-D LENGTHS
  567  IF (DHH/(SMU(M)/MU(M))  .GT.  DY(J) )  GO TO 570
C     HITS SIDE FIRST
       DH(LMAX+1-L) = DHH/SMU(M)
       XH(LMAX+1-L) = HIT
       YH(LMAX+1-L) = DHH/(SMU(M)/MU(M)) - DY(J)/2.
       GO TO 590
C     HITS BOTTOM FIRST
  570  DH(LMAX+1-L) = DY(J)/MU(M)
       HELP = DY(J) * (SMU(M)/MU(M))
       XH(LMAX+1-L) = SQRT ( A*A + HELP*HELP - 2.*A*HELP*COSPHI(L) )
       YH(LMAX+1-L) = DY(J)/2.
  590  CONTINUE
       IF (DBGPRT .LT. 1.E-20)  GO TO 1000
       LIM1P = LMAX +1 -LIM2
       LIM2P = LMAX +1 -LIM1
       WRITE (6,8005)
       WRITE (6,8001)  (L,DH(L),L=LIM1P,LIM2P)
       WRITE (6,8003)
       WRITE (6,8001)  (L,XH(L),L=LIM1P,LIM2P)
       WRITE (6,8006)
       WRITE (6,8001)  (L,YH(L),L=LIM1P,LIM2P)
 1000  RETURN
C
 8000  FORMAT (7H1FOR M=,I2,3H I=,I2,3H J=,I2//50H0PLANE CHARACTERISTIC L
```

```
     .ENGTHS FOR VERTICAL SURFACE/)
8001 FORMAT (3H L=,I2,8H LENGTH=,1PE14.6)
8002 FORMAT (1H0/44H CHARACTERISTIC LENGTHS FOR VERTICAL SURFACE/)
8003 FORMAT (1H0/30H X-COORDINATE OF INTERSECTIONS/)
8004 FORMAT (1H0//52H PLANE CHARACTERISITC LENGTHS FOR HORIZONTAL SURFA
     .CE/)
8005 FORMAT (1H0/46H CHARACTERISITC LENGTHS FOR HORIZONTAL SURFACE/)
8006 FORMAT (1H0/30H T-COORDINATE OF INTERSECTIONS/)
     END
```

Function JO(J, I, L, M)

```
      REAL  FUNCTION  JO(J,I,L,M)
C     JO(J,I,L,M) = RATE AT WHICH ENERGY IS PASSING INTO CELL (I,J) THRU
C       THE OUTSIDE SURFACE OR INTO CELL (I+1,J) THRU THE INSIDE SURFACE
C       IN FAN L IN DIRECTION MU(M) IN THE CASE OF ISOTROPIC RADIATION ,
C       ROSS=S=0           (DIVIDED BY I0)
C       ****************************************************************DHA8007
C          H E C T I C   C O M M O N                                    DHA8008
C       ****************************************************************DHA8009
       COMMON  AHN,    AHSIGN, AHN,    ATOM,   BCATAG, BCBTAG, BCLTAG,   HECCOM01
      1BCRTAG, CAPIN,  CAPS,   COUT,   CMXK,   CMXOM,  CNUE,   CO,       HECCOM02
      2COL,    CSTOP,  CCVV,   CYCLE,  DBGPRT, DGHEY,  DHNU,   DIANTP,   HECCOM03
      3DMIN,   DNN,    DT,     DTBUGH, DTC,    DTH,    DTNA,   DTQ,      HECCOM04
      4DTR,    DTR1,   DTR2,   DTUF,   DTVF,   DUMPT7, ECK,    EFRAC,    HECCOM05
      5EII,    ERRCRT, ETH,    EZERO,  FDTC,   FFA,    FFB,    FIFT,     HECCOM06
      6FUDGE,  GV,     HCB,    HCP,    HH,     HNU,    HNUP,   HNUS,     HECCOM07
      7HVB,    I,      IC,     IGOTO,  IH,     IHNU,   IMAX,   INPTAG,   HECCOM08
      8ISSSS,  ISR,    ITAG,   ITRMAX, IU,     IV,     I1,     I2,       HECCOM09
      9J,      JC,     JH,     JMAX,   JMAXA,  JU,     JV,     J5        HECCOM10
       COMMON  K,      KMAXA,  KMN1,   KMN2,   KP,     LLLL,   MMMM,     HECCOM11
      1MERGE,  MFTAG,  MZ,     N,      NC,     NHNU,   NK,     NPC,      HECCOM12
      2NM,     NRM,    NT,     NTAPE,  NY,     N1,     N2,     N3,       HECCOM13
      3NN,     ODOC,   PABOVE, PBLO,   PI,     PIDTS,  PRINTL, PRINTS,   HECCOM14
      4PROB,   PRR,    P0,     RADE,   RC,     RFT,    RPTAG,  RR,       HECCOM15
      5SCUR,   SCR,    SCHE,   SCYCLE, SGNL,   SIG,    SIGH,   SLUG,     HECCOM16
      6SN,     SPROB,  SVMAX,  SVS,    S1,     S2,     S3,     S4,       HECCOM17
      7SS,     T,      TAUUTS, THICK,  TMAX,   TRAD,   URH,    UT,       HECCOM18
      8UU,     UVMAX,  VABOVE, VAPE,   VBLO,   VEL,    WS,     WSA,      HECCOM19
      9XMAXX,          BLANK(49)                                         HECCOM20
       COMMON          AIX(1200),      AMX(1200),      ALAMH(1200),
      1ALAMV(1200),    B(1200),        CAP(1200),      ER(1200),
      2FIOUT(1200),    KFIT(1200),     P(1200),        SMLQ(1200),
      3THETA(1200),    U(1200),        V(1200),        XALP(1200),
      4XB(1200),       SOLID(400),     PL(200),        Y(101),
      5DT(100),        GAMC(100),      YAMC(100),      UL(100),          HECCOM26
      6W2(100),        X(53),          DX(52),         TAU(52),          HECCOM27
      7DETAB(50),      TEMP(12),       HEAD(12)                          HECCOM28
C                                                                        DHA8038
C     THE FOLLOWING VARIABLES IN HECTIC COMMON HAVE BEEN RENAMED IN THIS
C     ROUTINE  --  CV TO CCVV , ISEND TO ISSSS , L TO LLLL , M TO MMMM,
C     AND XMAX TO XMAXX
C       ****************************************************************DRA8039
C                                                                        DRA8040
       DIMENSION       FLEFT(100),     PR(100),        SIGC(100)         DRA8041
       EQUIVALENCE     (FLEFT,UL),     (PL,PR)
       EQUIVALENCE    (PL(101),SIGC)
       EQUIVALENCE (S20,BLANK(12)), (ZP120,BLANK(13)), (ZP136,BLANK(14)),
      1(KMX,BLANK(15)),(WT,BLANK(16)),(ANUMBR,BLANK(19)),(LOCZ,BLANK(22))
      2,(H,BLANK(25)),(NA,BLANK(26))
       EQUIVALENCE (BACC,BLANK(47)),(SACC,BLANK(48)),(TACC,BLANK(49))
C                                                                        DRA8045
       REAL  MERGE                                                       DRA8046
C                                                                        DRA8047
C                                                                        DRA8048
       COMMON   /RADI/    IAID,    KMAX,    ICHECK,  ISEND,              RAD1
      1L2,      L2P1,     LMAX,    M2,      MUMAX,   ISENDD,             RAD2
      2ISWEEP,  LLOW,     LHI                                            RAD
```

```
      COMMON      /RAD2/    ANGL1,    ANGL2,    ASN,      CHKSUM,         RAD3
     1DELTY,      FORPI,    HELP1,    HELP2,    LMDA,     PP,             RAD4
     2PI2,        PI4,      SINP2,    TANN,     TOLER,    [illegible]     RAD5
     3XMAX,       STEF                                                    RAD6
      COMMON      /RAD3/              DUM1(1),            AA(16),         RAD7
     1            COSPHI(6),          CVIN(6),            CVOUT(6),       RAD8
     2            CH(6),              MU(6),              PHI(6),         RAD9
     3            DUM2(1),            SINLP(6),           SMU(6),         RAD10
     4            W(6),               [illegible]V(6)                     RAD11
      COMMON      /INDEX/   M,L,LP                                        RAD12
      COMMON      /DRUMM/   LCHECK,   KK2,      NOBUFF                    RAD13
      COMMON      /DIS/     DV(6),    XV(6),    YV(6),    DH(6),          RAD14
     1            XH(6),    YH(6)                                         RAD15
      COMMON      /SUMCHK/            SUMB(50,6),         SUMO(50,6)      RAD16
      COMMON      /DNUT/    UTHETA(50),       RTHETA(50),    ATHETA(50)   RAD17
C
      REAL    MU , LMDA
C
      EQUIVALENCE   (FIOUT,OLDTH) , (U,ROSS) ,
     .   (CAP(1),BUFF1(2),BUFF(1),NOI) , (P(1),PLANCK(1),BUFF2(1),NOO)
      DIMENSION   OLDTH(1) , ROSS(1) , BUFF1(1) , BUFF(1) , PLANCK(1) ,
     .  BUFF2(1)
C
      EQUIVALENCE   (V(1),XA(1),EB(1))
      DIMENSION   XA(1)
      DIMENSION         EB(50,6) , EV(50,6) , ET(50,6) ,
     .          ETAIO(6,6) , ETAIT(6,6) , ETABI(6,6) , ETABT(6,6),
     .  ETABO(6,6) , ETAOI(6,6) , ETAOT(6,6) , ETAOO(6,6)
C
C
C
      JO    = DY(J) * 2. * SINP2 * ABS(COSPHI(L))
     .  * FORPI * X(I) * W(M) * SMU(M)
      RETURN
      END
```

Function JB(J, I, L, M)

```
      REAL FUNCTION  JB(J,I,L,M)
C     JB(J,I,L,M) = RATE AT WHICH ENERGY IS PASSING INTO CELL (I,J+1) THRU
C       THE BOTTOM SURFACE OR INTO CELL (I,J)   THRU THE TOP SURFACE
C       IN FAN L (MEASURED AT X(I)) IN DIRECTION MU(M) IN THE CASE OF
C       ISOTROPIC RADIATION , ROSS=S=0       (DIVIDED BY I0)
C
      REAL   LIM1 , LIM2
      REAL   I1
      REAL    MU , LMUA
      REAL  MERGE                                                       DRA8046
C
C     ******************************************************************DRA8007
C         H E C T I C   C O M M O N                                     DRA8008
C     ******************************************************************DRA8009
      COMMON  AHN,     AMBIEN, ANN,    ATOM,    BCATAG, BCBTAG, BCLTAG,   HECCOM01
     1UCHTAG, CAPIN,   CAPS,   COU(,   CMKK,    CMXON,  CNDE,   CO,       HECCOM02
     2COL,    CSTOP,   CCVV,   CTCLE,  DBGPRT,  DGNLY,  DHNU,   DIANTP,   HECCOM03
     3DMIN,   DNN,     DT,     DTHUGR, DTC,     DTH,    DTNA,   DTQ,      HECCOM04
     4DTR,    DTR1,    DTR2,   DTUP,   DTVF,    DUMPTT, ECK,    EFRAC,    HECCOM05
     5EII,    ERRCRT,  ETH,    EZERO,  FDIC,    FFA,    FFB,    FIFT,     HECCOM06
     6FUDGE,  GV,      HCB,    HCP,    HH,      HNU,    HNUP,   HNUS,     HECCOM07
     7HVB,    I,       IC,     IGOTO,  IH,      IHNU,   IMAX,   IMPTAG,   HECCOM08
     8ISSSS,  ISR,     ITAG,   ITRMAX, IU,      IV,     IIII,   I2,       HECCOM09
     9J,      JC,      JH,     JMAX,   JMAXA,   JU,     JV,     J5        HECCOM10
      COMMON  K,       KMAXA,  KMN1,   KMN2,    KP,     LLLL,   MMMM,     HECCOM11
     1MERGE,  MFTAG,   MZ,     N,      NC,      NHNU,   NK,     NPC,      HECCOM12
     2NR,     NRM,     NT,     NTAPE,  NY,      N1,     N2,     N3,       HECCOM13
     3NN,     ODDC,    PABOVE, PBLO,   PI,      PIDTS,  PRINTL, PRINTS,   HECCOM14
     4PROB,   PRR,     P,      RADE,   RC,      RFT,    RPTAG,  RR,       HECCOM15
     5SCOR,   SCR,     SCRE,   SCTCLE, SGNL,    SIG,    SIGH,   SLUG,     HECCOM16
     6SN,     SPROB,   SVMAX,  SVS,    S1,      S2,     S3,     S4,       HECCOM17
     7SS,     T,       TAUOTS, THICK,  TMAX,    TRAD,   URR,    UT,       HECCOM18
     8UU,     UVMAX,   VABOVE, VAPE,   VBLO,    VEL,    WS,     WSA,      HECCOM19
     9XMAXX,           BLANK(49)                                          HECCOM20
      COMMON           AIX(1200),       AMX(1200),       ALAMH(1200),
     1ALAMV(1200),     B(1200),         CAP(1200),       ER(1200),
     2FIOUT(1200),     KFIT(1200),      P(1200),         SMLQ(1200),
     3THETA(1200),     U(1200),         V(1200),         XALP(1200),
     4XB(1200),        SOLID(400),      PL(200),         Y(101),
     5DT(100),         GAMC(100),       YAMC(100),       UL(100),         HECCOM26
     6B2(100),         X(53),           DX(52),          TAU(52),         HECCOM27
     7BETAB(50),       TEMP(12),        HEAD(12)                          HECCOM28
C                                                                       DRA8038
C     THE FOLLOWING VARIABLES IN HECTIC COMMON HAVE BEEN RENAMED IN THIS
C     ROUTINE  --  CV TO CCVV , ISEND TO ISSSS , L TO LLLL , M TO MMMM,
C     XMAX TO XMAXX , AND I1 TO IIII
C       ****************************************************************DRA8039
C                                                                       DRA8040
       DIMENSION         FLEFT(100),       PR(100),         SIGC(100)     DRA8041
       COMMON     /RAD1/     IAID,     KMAX,     ICHECK,    ISEND,      RAD1
      1L2,        L2P1,      LMAX,     M2,       MUMAX,     ISENDD,     RAD2
      2ISWEEP,    LLOW,      LHI                                        RAD
       COMMON     /RAD2/     ANGL1,    ANGL2,    ASN,       CHKSUM,     RAD3
      1DELTT,     FORPI,     HELP1,    HELP2,    LMDA,      PP,         RAD4
      2PI2,       PI4,       SINP2,    TANN,     TOLER,     XMIN,       RAD5
      3XMAX,      STEF                                                  RAD6
```

```
      COMMON    /RADS/                  DUM1(1),              AA(16),        RAD7
     1          COSPHI(6),              CVIN(6),              CVOUT(6),      RAD8
     2          CH(6),                  MU(6),                PHI(6),        RAD9
     3          DUM2(1),                SINLP(6),             SMU(6),        RAD10
     4          W(6),                   CV(6)                                RAD11
      COMMON   /INDEX/   M,L,LP                                              RAD12
      COMMON    /DRUMM/   LCHECK,   KK2,       NOBUFF                        RAD13
      COMMON    /DIS/     DV(6),    XV(6),     YV(6),     DH(6),      ,      RAD14
     1          XH(6),    YH(6)                                              RAD15
      COMMON    /SUMCHK/            SUMB(50,6),           SUMO(50,6)         RAD16
      COMMON    /BNDT/    BTHETA(50),      RTHETA(50),    ATHETA(50)         RAD17
C
C
      DIMENSION   OLDTH(1) , ROSS(1) , BUFF1(1) , BUFF(1) , PLANCK(1) ,
     .  BUFF2(1)
C
      DIMENSION   XA(1)
      DIMENSION             EB(50,6) , EV(50,6) , ET(50,6) ,
     .          ETAIO(6,6) , ETAIT(6,6) , ETABI(6,6) , ETABT(6,6),
     .  ETABO(6,6) , ETAOI(5,6) , ETAOT(6,6) , ETAOO(6,6)
C                                                                     DRA8047
C
      EQUIVALENCE      (FLEFT,UL),      (PL,PR)
      EQUIVALENCE     (PL(101),SIGC)
      EQUIVALENCE (S20,BLANK(12)), (ZP126,BLANK(13)), (ZP136,BLANK(14)),
     1(KMX,BLANK(15)),(WT,BLANK(16)),(ANUMBR,BLANK(19)),(LOCZ,BLANK(22))
     2,(H,BLANK(25)),(NA,BLANK(26))
      EQUIVALENCE (BACC,BLANK(47)),(SACC,BLANK(48)),(TACC,BLANK(49))
C                                                                     DRA8045
C                                                                     DRA8048
      EQUIVALENCE   (F1OUT,OLDTH) , (U,ROSS) ,
     .   (CAP(1),BUFF1(1),BUFF(1),NOI) , (P(1),PLANCK(1),BUFF2(1),NOO)
      EQUIVALENCE   (V(1),XA(1),EB(1))
C
      I1(A,R) = .5*( R*R*ASIN(A/R) + A*SQRT(R*R-A*A) )
C
      LL = L
      IF (L .GT. L2)  LL=LMAX+1-L
      LIM1 = AMAX1( X(I-1) , AA(LL-1) )
      LIM2 = AMAX1( X(I-1) , AA(LL) )
      JB = I1(AA(LL),X(I)) - I1(AA(LL),LIM2) - I1(AA(LL-1),X(I))
     . + I1(AA(LL-1),LIM1) + PI4*(LIM2*LIM2 - LIM1*LIM1)
      JB = JB * FORP1 * W(M) * ABS(MU(M))
      RETURN
      END
```

Function SINT(A1, B1, EPS1, FUNC)

```
      FUNCTION SINT(A1,B1,EPS1,FUNC)
C     ------------------------------------------------------------
C     THIS ROUTINE DOES SIMPSONS INTEGRATION
C     ALGORITHM 182     COMMUNICATIONS OF THE ACM
C     ------------------------------------------------------------
C     ------------------------------------------------------------
C     A IS THE LOWER LIMIT OF INTEGRATION
C     B IS THE UPPER LIMIT OF INTEGRATION
C     EPS IS USED IN THE ACCURACY CRITERION
C     FUNC IS THE EVALUATING FUNCTION NAME
C     ------------------------------------------------------------
C     ------------------------------------------------------------
      DIMENSION DX(30),EPSP(30),X2(30),X3(30),F2(30),F3(30),F4(30),
     1 FMP(30),FBP(30),EST2(30),EST3(30),LR(30),PVAL(30,3)
      A = A1
      B = B1
      EPS = EPS1
      L = 0
      ABR = 1.0
      I30 = 0
      EST = 1.0
      DA = B-A
      FA = FUNC(A)
      FM = FUNC((A + B)/2.0)*4.0
      FB = FUNC(B)
  110 L = L+1
      DX(L) = DA/3.0
      SX = DX(L)/6.0
      F1 = 4.0*FUNC((A+DX(L)/2.0))
      X2(L) = A + DX(L)
      F2(L) = FUNC(X2(L))
      X3(L) = X2(L)+ DX(L)
      F3(L) = FUNC(X3(L))
      EPSP(L) = EPS
      F4(L) = 4.0 * FUNC(X3(L) + DX(L)/2.0)
      FMP(L) = FM
      EST1 = (FA + F1 + F2(L)) * SX
      FBP(L) = FB
      EST2(L) = (F2(L) + F3(L) + FM) * SX
      EST3(L) = (F3(L) + F4(L) + FB) * SX
      SUM = EST1 + EST2(L) + EST3(L)
      IF ( ABS(EST-SUM) .LT. EPSP(L)    .AND.    L .NE. 1 )  GO TO 10
      IF (L .GE. 30)  GO TO 15
    1 LR(L) = 1
      DA = DX(L)
      FM = F1
      FB = F2(L)
      EPS = EPSP(L)/1.7
      EST = EST1
      GO TO 110
    2 LR(L) = 2
      DA = DX(L)
      FA = F2(L)
      FM = FMP(L)
      FB = F3(L)
      EPS = EPSP(L)/1.7
      EST = EST2(L)
      A = X2(L)
```

```
      GO TO 110
    3 LR(L) = 3
      DA = UX(L)
      FA = F3(L)
      FM = F4(L)
      FB = FWP(L)
      EPS = EPSP(L)/1.7
      EST = EST3(L)
      A = X3(L)
      GO TO 110
    4 SUM = PVAL(L,1) + PVAL(L,2) + PVAL(L,3)
      IF(L .GT.1)GO TO 10
      SINT = SUM
      RETURN
C     DONE AT THIS LEVEL
   10 L = L-1
      LTO = LR(L)
      PVAL(L,LTO) = SUM
      GO TO (2,3,4), LTO
   15 IF (I30 .EQ. 1)  GO TO 10
      I30 = 1
      WRITE (6,100)
  100 FORMAT (22H SINT REACHED LEVEL 30)
      GO TO 10
      END
```

Function IN(Z)

```
      REAL FUNCTION IN(Z)
C
C     ROUTINE TO CALCULATE -SINT- INTEGRANDS FOR RADIATION ENTERING THE
C     INSIDE CELL WALL AT Y=T(J)+Z
C
C     *****************************************************************DRA8007
C         H E C T I C   C O M M O N                                     DRA8008
C     *****************************************************************DRA8009
      COMMON  AHN,    AMBIEN, ANN,    ATOM,   BCATAG, BCBTAG, BCLTAG,   HECCOM01
     1BCRTAG, CAPIN,  CAPS,   CDUT,   CMXK,   CMXOM,  CNDE,   CO,       HECCOM02
     2COE,    CSTOP,  CCVV,   CYCLE,  DBGPRT, DGHEY,  DHNU,   DIANTP,   HECCOM03
     3DMIN,   DNN,    DT,     DTBUGR, DTC,    DTH,    DTNA,   DTQ,      HECCOM04
     4DTH,    DTR1,   DTR2,   DTUF,   DTVF,   DUMPT7, ECK,    EFRAC,    HECCOM05
     5E11,    ERRCRT, ETH,    EZERO,  FDIC,   FFA,    FFB,    FIFT,     HECCOM06
     6FUDGE,  GV,     HCB,    HCP,    HH,     HNU,    HNUP,   HNUS,     HECCOM07
     7HVB,    I,      IC,     IGOTO,  IH,     IHNU,   IMAX,   IMPTAG,   HECCOM08
     8ISSSS,  ISR,    ITAG,   ITRMAX, IU,     IV,     I1,     I2,       HECCOM09
     9J,      JC,     JH,     JMAX,   JMAXA,  JU,     JV,     J5        HECCOM10
      COMMON  K,      KMAXA,  KMN1,   KMN2,   KP,     LLLL,   MMMM,     HECCOM11
     1MERGE,  MFTAG,  MZ,     N,      NC,     NHNU,   NK,     NPC,      HECCOM12
     2NR,     NRM,    NT,     NTAPE,  NY,     N1,     N2,     N3,       HECCOM13
     3NN,     ODOC,   PABOVE, PBLO,   PI,     PIDTS,  PRINTL, PRINTS,   HECCOM14
     4PROB,   PRR,    PW,     RADE,   RC,     RFT,    RPTAG,  RR,       HECCOM15
     5SCUR,   SCR,    SCRE,   SCYCLE, SGNL,   SIG,    SIGH,   SLUG,     HECCOM16
     6SH,     SPROB,  SVMAX,  SVS,    S1,     S2,     S3,     S4,       HECCOM17
     7SS,     T,      TAUDTS, THICK,  TMAX,   TRAD,   URR,    UT,       HECCOM18
     8UU,     UVMAX,  VABOVE, VAPE,   VBLO,   VEL,    WS,     WSA,      HECCOM19
     9XMAXX,          BLANK(49)                                         HECCOM20
      COMMON          A1X(1200),      AMX(1200),      ALAMH(1200),
     1ALAMV(1200),    B(1200),        CAP(1200),      ER(1200),
     2FIOUT(1200),    KFIT(1200),     P(1200),        SMLQ(1200),
     3THETA(1200),    U(1200),        V(1200),        XALP(1200),
     4XB(1200),       SOLID(400),     PL(200),        Y(101),
     5DY(100),        GAMC(100),      YAMC(100),      UL(100),          HECCOM26
     6B2(100),        X(53),          DX(52),         TAU(52),          HECCOM27
     7BETAB(50),      TEMP(12),       HEAD(12)                          HECCOM28
C                                                                       DRA8038
C     THE FOLLOWING VARIABLES IN HECTIC COMMON HAVE BEEN RENAMED IN THIS
C     ROUTINE  --  CV TO CCVV , ISEND TO ISSSS , L TO LLLL , M TO MMMM,
C     AND XMAX TO XMAXX
C     *****************************************************************DRA8039
C                                                                       DRA8040
      DIMENSION       FLEFT(100),     PR(100),        SIGC(100)         DRA8041
      EQUIVALENCE     (FLEFT,UL),     (PL,PR)
      EQUIVALENCE     (PL(101),SIGC)
      EQUIVALENCE (S29,BLANK(12)), (ZP126,BLANK(13)), (ZP136,BLANK(14)),
     1(KMX,BLANK(15)),(NT,BLANK(16)),(ANUMBR,BLANK(19)),(LOCZ,BLANK(22))
     2,(H,BLANK(25)),(NA,BLANK(26))
      EQUIVALENCE (BACC,BLANK(47)),(SACC,BLANK(48)),(TACC,BLANK(49))
C                                                                       DRA8045
      REAL  MERGE                                                       DRA8046
C                                                                       DRA8047
C                                                                       DRA8048
      COMMON  /RAD1/   IAID,    KMAX,    ICHECK,  ISEND,                RAD1
     1L2,      L2P1,    LMAX,    M2,      MUMAX,   ISENDD,              RAD2
     2ISWEEP,  LLOW,    LHI                                             RAD
```

```
      COMMON     /RAD2/     ANGL1,     ANGL2,     ASN,       CHKSUM,         RAD3
     1DELTY,     FORPI,     HELP1,     HELP2,     LMDA,      PP,             RAD4
     2PI2,       PI4,       SINP2,     TANN,      TOLER,     XMIN,           RAD5
     3XMAX,      STEF                                                        RAD6
      COMMON     /RAD3/                DUM1(1),              AA(16),         RAD7
     1           COSPHI(6),            CVIN(6),              CVOUT(6),       RAD8
     2           CH(6),                MU(6),                PHI(6),         RAD9
     3           DUM2(1),              SINLP(6),             SMU(6),         RAD10
     4           W(6),                 CV(6)                                 RAD11
      COMMON    /INDEX/   M,L,LP                                             RAD12
      COMMON     /DRUMM/    LCHECK,    KK2,       NOBUFF                     RAD13
      COMMON     /DIS/      DV(6),     XV(6),     YV(6),     DH(6),          RAD14
     1           XH(6),     YH(6)                                            RAD15
      COMMON     /SUMCHK/              SUMB(50,6),           SUMO(50,6)      RAD16
      COMMON     /BNDT/     BTHETA(50),       RTHETA(50),    ATHETA(50)      RAD17
C
      REAL    MU , LMDA
C
      EQUIVALENCE    (FIOUT,OLDTH) , (U,ROSS) ,
     .   (CAP(1),BUFF1(1),BUFF(1),NOI) , (P(1),PLANCK(1),BUFF2(1),NOO)
      DIMENSION   OLDTH(1) , ROSS(1) , BUFF1(1) , BUFF(1) , PLANCK(1) ,
     .  BUFF2(1)
C
      EQUIVALENCE    (V(1),XA(1),EB(1))
      DIMENSION   XA(1)
      DIMENSION              EB(50,6) , EV(50,6) , ET(50,6) ,
     .            ETAIO(6,6) , ETAIT(6,6) , ETABI(6,6) , ETABT(6,6),
     .  ETABO(6,6) , ETAOI(6,6) , ETAOT(6,6) , ETAOO(6,6)
C
C
C
C   CALCULATE GAMMA(I-1)
      LMDA = TANN * (DELTY-Z)
      IF (XMAX .GT. XMIN+LMDA) GO TO 10
      IF (LMDA .GT. XMAX+XMIN) GO TO 20
      GAM  = AMAX1( PI2 , ACOS( (LMDA*LMDA+HELP1)/(2.*XMIN*LMDA) ) )
      GO TO 30
 10   GAM  = PI
      GO TO 30
 20   GAM  = PI2
C   INTEGRAND
 30   IN = SIN( AMAX1(ANGL1, GAM ) )
      RETURN
      END
```

Function BOT(Z)

```
      FUNCTION BOT(Z)
C     ******************************************************************DRA8007
C        H E C T I C   C O M M O N                                      DRA8008
C     ******************************************************************DRA8009
       COMMON  AHN,    AMBIEN, ANN,    ATOM,   BCATAG, BCBTAG, BCLTAG,  HECCOM01
     1BCRTAG,  CAPIN,  CAPS,   COUT,   CMXK,   CMXOM,  CNDE,   CO,      HECCOM02
     2COE,     CSTOP,  CCVV,   CYCLE,  DBGPRT, DGRET,  DHNU,   DIANTP,  HECCOM03
     3DMIN,    DNN,    DT,     DTBUGR, DTC,    DTH,    DTNA,   DTQ,     HECCOM04
     4DIR,     DTR1,   DTR2,   DTUF,   DTVF,   DUMPT7, ECK,    EFRAC,   HECCOM05
     5EII,     ERRCRT, LTH,    EZERO,  FDTC,   FFA,    FFB,    F1FT,    HECCOM06
     6FUDGE,   GV,     HCB,    HCP,    HH,     HNU,    HNUP,   HNUS,    HECCOM07
     7HVB,     I,      IC,     IGOTO,  IH,     IHNU,   IMAX,   IMPTAG,  HECCOM08
     8ISSSS,   ISR,    ITAG,   ITRMAX, IU,     IV,     I1,     I2,      HECCOM09
     9J,       JC,     JH,     JMAA,   JMAXA,  JU,     JV,     J5       HECCOM10
       COMMON  K,      KMAXA,  KMN1,   KMN2,   KP,     LLLL,   MMMM,    HECCOM11
     1MERGE,   MFTAG,  MZ,     N,      NC,     NHNU,   NK,     NPC,     HECCOM12
     2NR,      NRM,    NT,     NTAPE,  NY,     N1,     N2,     N3,      HECCOM13
     3N4,      ODDC,   PABOVE, PBLO,   P1,     P1DTS,  PRINTL, PRINTS,  HECCOM14
     4PROB,    PRR,    PW,     RADE,   RC,     RFT,    RPTAG,  RR,      HECCOM15
     5SCDR,    SCR,    SCRE,   SCYCLE, SGNL,   SIG,    SIGH,   SLUG,    HECCOM16
     6SN,      SPROB,  SVMAX,  SVS,    S1,     S2,     S3,     S4,      HECCOM17
     7SS,      T,      TAUDTS, THICK,  TMAX,   TRAD,   URR,    UT,      HECCOM18
     8UU,      UVMAX,  VABOVE, VAPE,   VBLO,   VEL,    WS,     WSA,     HECCOM19
     9XMAXX,           BLANK(49)                                        HECCOM20
       COMMON          A1X(1200),      AMX(1200),      ALAMH(1200),
     1ALAMV(1200),     B(1200),        CAP(1200),      ER(1200),
     2FIOUT(1200),     KFIT(1200),     P(1200),        SMLQ(1200),
     3THETA(1200),     U(1200),        V(1200),        XALP(1200),
     4XB(1200),        SOLID(400),     PL(200),        Y(101),
     5DY(100),         GAMC(100),      YAMC(100),      UL(100),         HECCOM26
     6W2(100),         X(53),          DX(52),         TAU(52),         HECCOM27
     7BETAB(50),       TEMP(12),       HEAD(12)                         HECCOM28
C                                                                       DRA8038
C    THE FOLLOWING VARIABLES IN HECTIC COMMON HAVE BEEN RENAMED IN THIS
C    ROUTINE  --  CV TO CCVV , ISEND TO ISSSS , L TO LLLL , M TO MMMM,
C    AND XMAX TO XMAXX
C     ******************************************************************DRA8039
C                                                                       DRA8040
      DIMENSION        FLEFT(100),     PR(100),        SIGC(100)        DRA8041
      EQUIVALENCE      (FLEFT,UL),     (PL,PR)
      EQUIVALENCE     (PL(101),SIGC)
      EQUIVALENCE (S20,BLANK(12)), (ZP126,BLANK(13)), (ZP136,BLANK(14)),
     1(KMX,BLANK(15)),(WT,BLANK(16)),(ANUMBR,BLANK(19)),(LOCZ,BLANK(22))
     2,(H,BLANK(25)),(NA,BLANK(26))
      EQUIVALENCE (BACC,BLANK(47)),(SACC,BLANK(48)),(TACC,BLANK(49))
C                                                                       DRA8045
      REAL  MERGE                                                       DRA8046
C                                                                       DRA8047
C                                                                       DRA8048
      COMMON   /RAD1/    IAIU,     KMAX,     ICHECK,   ISEND,           RAD1
     1L2,      L2P1,     LMAX,     M2,       MUMAX,    ISENDD,          RAD2
     2ISWEEP,  LLOW,     LHI                                            RAD
      COMMON   /RAD2/    ANGL1,    ANGL2,    ASN,      CHKSUM,          RAD3
     1DELTY,   FORPI,    HELP1,    HELP2,    LMDA,     PP,              RAD4
     2P12,     P14,      SINP2,    TANN,     TOLER,    XMIN,            RAD5
     3XMAX,    STEF                                                     RAD6
```

```
      COMMON     /RAD3/                 DUM1(1),            AA(16),          RAD7
     1           COSPHI(6),             CVIN(6),            CVOUT(6),        RAD8
     2           CH(6),                 MU(6),              PHI(6),          RAD9
     3           DUM2(1),               SINLP(6),           SMU(6),          RAD10
     4           W(6),                  CV(6)                                RAD11
      COMMON    /INDEX/   M,L,LP                                             RAD12
      COMMON     /DHUMM/   LCHECK,  KK2,     NOBUFF                          RAD13
      COMMON     /DIS/     UV(6),   XV(6),   YV(6),     DH(6),               RAD14
     1           XH(6),    YH(6)                                             RAD15
      COMMON     /SUMCHK/               SUMB(50,6),         SUMO(50,6)       RAD16
      COMMON     /BNDT/    BTHETA(50),         RTHETA(50),     ATHETA(50)    RAD17
C
      REAL    MU , LMDA
C
      EQUIVALENCE    (FIOUT,OLDTH) , (U,ROSS) ,
     .    (CAP(1),BUFF1(1),BUFF(1),NOI) , (P(1),PLANCK(1),BUFF2(1),NOO)
      DIMENSION   OLDTH(1) , ROSS(1) , BUFF1(1) , BUFF(1) , PLANCK(1) ,
     .  BUFF2(1)
C
      EQUIVALENCE    (V(1),XA(1),EB(1))
      DIMENSION   XA(1)
      DIMENSION           EB(50,6) , EV(50,6) , ET(50,6) ,
     .            ETAIO(6,6) , ETAIT(6,6) , ETABI(6,6) , ETABT(6,6),
     .  ETABO(6,6) , ETAOI(6,6) , ETAOT(6,6) , ETAOO(6,6)
C
C
C
      GO TO (10,20) , ISEND
C
C    FUNCTION FOR ETABI
C
 10   IF (Z.LT.XMIN+LMDA  .AND.  I.NE.1)  GO TO 11
      GAM = 0.
      GO TO 17
 11   IF (LMDA .GT. SQRT(Z*Z-XMIN*XMIN))  GO TO 15
      GAM = ACOS( (HELP1 + Z*Z) / (2. * Z * LMDA ) )
      GO TO 17
 15   GAM = ASIN(XMIN/Z)
 17   BOT = Z*( AMIN1(ASIN(ANGL1/Z),GAM) )
      GO TO 100
C
C    FUNCTIONS FOR ETABT AND ETABO
C
C    CALCULATE GAMMAS
C     LOWER GAMMA
 20   IF ( Z.LT.(XMIN+LMDA) .AND. I.NE.1 ) GO TO 22
      GAM1 = 0.
      GO TO 26
 22   IF ( LMDA .GT. SQRT(Z*Z-XMIN*XMIN) ) GO TO 25
      GAM1 = ACOS( (HELP1+Z*Z)/(2.*Z*LMDA) )
      GO TO 26
 25   GAM1 = ASIN(XMIN/Z)
C     UPPER GAMMA
 26   IF (Z .LE. XMAX-LMDA)  GO TO 27
      IF (Z .LE. LMDA-XMAX)  GO TO 28
```

```
      HELP = (HELP2 + Z*Z)/(2.*Z*LMDA)
      IF (HELP .LT. 0.)  HELP=AMAX1(HELP,-1.)
      IF (HELP .GT. 0.)  HELP=AMIN1(HELP,1.)
      GAM = ACOS(HELP)
      GAM2 = AMAX1(GAM,GAM1)
      GO TO 29
 27   GAM2 = PI
      GO TO 29
 28   GAM2 = GAM1
C
 29   GO TO (23,24,31,40,50) , IAIO
C   ETABT
 23   AAB = ACOS(LMDA/Z)
      A = AMIN1(AAB,GAM2,ASIN( AMIN1(ANGL1/Z,1.)) )
      B = AMAX1( GAM1 , ASIN(ANGL2/Z) )
      GO TO 21
 24   AAB = ACOS( AMIN1(LMDA/Z,1.) )
      A = AMIN1 ( GAM2 , ASIN( AMIN1(1.,ANGL1/Z)) )
      B = AMAX1( ASIN(ANGL2/Z),GAM1,AAB )
      GO TO 21
 31   A = AMIN1( (PI-ASIN(ANGL1/Z)), GAM2)
      B = (PI - ASIN( AMIN1(ANGL2/Z,1.)))
 21   BOT = Z * AMAX1(A-B,0.)
      GO TO 100
C   ETABO
 40   A = ASIN( AMIN1(1.,ANGL1/Z) )
      B = AMAX1( GAM2 , ASIN(ANGL2/Z) )
      GO TO 61
 50   A = PI - ASIN( ANGL1/Z )
      B = AMAX1( GAM2 , (PI - ASIN( AMIN1(1.,ANGL2/Z)) )
 61   BOT = Z * AMAX1(A-B,0.)
 100  RETURN
      END
```

Function OUT(Z)

```
      FUNCTION   OUT(Z)
C
C     ROUTINE TO CALCULATE -SINT- INTEGRANDS FOR RADIATION ENTERING THE
C       OUTSIDE THE CELL WALL AT Y=Y(J)+Z
C       ****************************************************************DRA8007
C          H E C T I C   C O M M O N                                   DRA8008
C       ****************************************************************DRA8009
       COMMON  AHN,     AMBIEN, ANN,    ATOM,   BCATAG, BCBTAG, BCLTAG, HECCOM01
      1BCRTAG, CAPIN,   CAPS,   COUT,   CMXK,   CMXOM,  CNDE,   CO,     HECCOM02
      2COE,    CSTOP,   CCVV,   CYCLE,  DBGPRT, DGREY,  DHNU,   DIANTP, HECCOM03
      3DMIN,   DNN,     DT,     DTBUGR, DTC,    DTH,    DTNA,   DT0,    HECCOM04
      4DTR,    DTR1,    DTR2,   DTUF,   DTVF,   DUMPT7, ECK,    EFRAC,  HECCOM05
      5EII,    ERRCRT,  ETH,    EZERO,  FDTC,   FFA,    FFB,    FIFT,   HECCOM06
      6FUDGE,  GV,      HCB,    HCP,    HH,     HNU,    HNUP,   HNUS,   HECCOM07
      7HVB,    I,       IC,     IGOTO,  IH,     IHNU,   IMAX,   IMPTAG, HECCOM08
      8ISSSS,  ISR,     ITAG,   ITRMAX, IU,     IV,     I1,     I2,     HECCOM09
      9J,      JC,      JH,     JMAX,   JMAXA,  JU,     JV,     J5      HECCOM10
       COMMON  K,       KMAXA,  KNN1,   KNN2,   KP,     LLLL,   MMMM,   HECCOM11
      1MERGE,  MFTAG,   MZ,     N,      NC,     NHNU,   NK,     NPC,    HECCOM12
      2NM,     NRM,     NT,     NTAPE,  NY,     N1,     N2,     N3,     HECCOM13
      3NN,     OODC,    PABOVE, PBLO,   PI,     PIDTS,  PRINTL, PRINTS, HECCOM14
      4PROB,   PHR,     PW,     RADE,   RC,     RFT,    RPTAG,  RR,     HECCOM15
      5SCDR,   SCR,     SCRE,   SCYCLE, SGNL,   SIG,    SIGH,   SLUB,   HECCOM16
      6SH,     SPROB,   SVMAX,  SVS,    S1,     S2,     S3,     S4,     HECCOM17
      7TS,     T,       TAUDTS, THICK,  TMAX,   TRAD,   URR,    UT,     HECCOM18
      8UU,     UVMAX,   VABOVE, VAPE,   VBLO,   VEL,    WS,     WSA,    HECCOM19
      9XMAXX,           BLANK(49)                                       HECCOM20
       COMMON           AIX(1200),        AMX(1200),        ALAMH(1200),
      1ALAMV(1200),     B(1200),          CAP(1200),        ER(1200),
      2FIOUT(1200),     KFIT(1200),       P(1200),          SML0(1200),
      3THETA(1200),     U(1200),          V(1200),          XALP(1200),
      4XB(1200),        SOLIU(400),       PL(200),          Y(101),
      5UT(100),         GAMC(100),        YAMC(100),        UL(100),    HECCOM26
      6W2(100),         X(53),            UX(52),           TAU(52),    HECCOM27
      7WETAB(50),       TEMP(12),         HEAD(12)                      HECCOM28
C                                                                       DRA8038
C     THE FOLLOWING VARIABLES IN HECTIC COMMON HAVE BEEN RENAMED IN THIS
C     ROUTINE  --  CV TO CCVV , ISEND TO ISSSS , L TO LLLL , M TO MMMM,
C     AND XMAX TO XMAXX
C       ****************************************************************DRA8039
C                                                                       DRA8040
       DIMENSION        FLEFT(100),       PR(100),          SIGC(100)   DRA8041
       EQUIVALENCE      (FLEFT,UL),       (PL,PR)
       EQUIVALENCE     (PL(101),SIGC)
       EQUIVALENCE (S20,BLANK(12)), (ZP120,BLANK(13)), (ZP136,BLANK(14)),
      1(KMX,BLANK(15)),(WT,BLANK(16)),(ANUMBR,BLANK(19)),(LOCZ,BLANK(22))
      2,(H,BLANK(25)),(NA,BLANK(26))
       EQUIVALENCE (BACC,BLANK(47)),(SACC,BLANK(48)),(TACC,BLANK(49))
C                                                                       DRA8045
       REAL  MERGE                                                      DRA8046
C                                                                       DRA8047
C                                                                       DRA8048
       COMMON     /RAD1/     IAID,      KMAX,      ICHECK,    ISEND,    RAD1
      1L2,        L2P1,      LMAX,      M2,        NUMAX,     ISENDD,   RAD2
      2ISWEEP,    LLOW,      LH1                                        RAD
       COMMON     /RAD2/     ANGL1,     ANGL2,     ASN,       CHKSUM,   RAD3
```

```
     1DELTY,      FORPI,     HELP1,     HELP2,     LMDA,      PP,
     2PI2,        PI4,       SINP2,     TANN,      TOLER,     XMIN,
     3XMAX,       STEF
      COMMON     /RAD3/                DUM1(1),              AA(16),
     1            COSPHI(6),            CVIN(6),              CVOUT(6),
     2            CH(6),                MU(6),                PHI(6),
     3            DUM2(1),              SINLP(6),             SMU(6),
     4            W(6),                 CV(6)
      COMMON     /INDEX/    M,L,LP
      COMMON     /DRUMM/    LCHECK,    KK2,        NOBUFF
      COMMON     /DIS/      DV(6),     XV(6),      YV(6),      DH(6),
     1            XH(6),     YH(6)
      COMMON     /SUMCHK/              SUMB(50,6),            SUMO(50,6)
      COMMON     /BNDT/     BTHETA(50),        RTHETA(50),     ATHETA(50)
C
      REAL     MU , LMDA
C
      EQUIVALENCE    (FIOUT,OLDTH) , (U,ROSS) ,
     .    (CAP(1),BUFF1(1),BUFF(1),NOI) , (P(1),PLANCK(1),BUFF2(1),NOO)
      DIMENSION    OLDTH(1) , ROSS(1) , BUFF1(1) , BUFF(1) , PLANCK(1) ,
     .  BUFF2(1)
C
      EQUIVALENCE    (V(1),XA(1),EB(1))
      DIMENSION    XA(1)
      DIMENSION             EB(50,6) , EV(50,6) , ET(50,6) ,
     .             ETAIO(6,6) , ETAIT(6,6) , ETABI(6,6) , ETABT(6,6),
     .  ETABO(6,6) , ETAOI(6,6) , ETAOT(6,6) , ETAOO(6,6)
C
C
C
C
      GO TO (10,20,50) , ISEND
C
C   INTEGRAND FOR ETAOI
C
 10   OUT = SIN( AMIN1(ANGL1,GAMMO(1,Z)) )
      GO TO 70
C
C   INTEGRAND FOR ETAOT
C
 20   GO TO (30,40) , IAID
C   CASE1 -- EXIT INWARD
 30   A = AMIN1( ANGL2 , ACOS( (DELTY-Z)*TANN/XMAX) )
      B = AMAX1( ANGL1 , GAMMO(1,Z) )
      GO TO 60
C   CASE2 -- EXIT OUTWARD
 40   A = AMIN1( ANGL2 , GAMMO(2,Z) )
      B = AMAX1( ANGL1 , GAMMO(1,Z) , ACOS( AMIN1(1.,(DELTY-Z)*TANN
     .  /XMAX) ) )
      GO TO 60
C
C   INTEGRAND FOR ETAOO
C
 50   A = ANGL2
      B = AMAX1( ANGL1 , GAMMO(2,Z) )
```

```
60   OUT = AMAX1( 0. , SIN(A)-SIN(B) )
70   RETURN
     END
```

Function GAMMO(IG, Z)

```
      FUNCTION   GAMMO(IG,Z)
C     ROUTINE TO CALCULATE GAMMA(I) AND GAMMA(I-1) FOR RADIATION ENTERING
C       THE OUTSIDE CELL WALL AT Y=Y(J)+Z
C
C      ******************************************************************DRA8007
C          H E C T I C   C O M M O N                                     DRA8008
C      ******************************************************************DRA8009
       COMMON  AHN,    AMBIEN, ANN,    ATOM,   BCATAG, BCBTAG, BCLTAG,  HECCOM01
      1BCRTAG, CAPIN,  CAPS,   CDUT,   CMXK,   CMXOM,  CNDE,   CO,      HECCOM02
      2COL,    CSTOP,  CCVV,   CYCLE,  DBGPRT, DGREY,  DHNU,   DIANTP,  HECCOM03
      3DMIN,   DNN,    UT,     DTBUGR, DTC,    DTH,    DTNA,   DTQ,     HECCOM04
      4DTR,    DTR1,   DTR2,   DTUF,   DTVF,   DUMPT7, ECK,    EFRAC,   HECCOM05
      5EI1,    ERRCRT, ETH,    EZERO,  FDTC,   FFA,    FFB,    FIFT,    HECCOM06
      6FUDGE,  GV,     HCB,    HCP,    HH,     HNU,    HNUP,   HNUS,    HECCOM07
      7HVB,    I,      IC,     IGOTO,  IH,     IHNU,   IMAX,   IMPTAG,  HECCOM08
      8ISSSS,  ISR,    ITAG,   ITRMAX, IU,     IV,     I1,     I2,      HECCOM09
      9J,      JC,     JH,     JMAX,   JMAXA,  JU,     JV,     J5       HECCOM10
       COMMON  K,      KMAXA,  KMN1,   KMN2,   KP,     LLLL,   MMMM,    HECCOM11
      1MERGE,  MFTAG,  MZ,     N,      NC,     NHNU,   NK,     NPC,     HECCOM12
      2NR,     NRM,    NT,     NTAPE,  NT,     N1,     N2,     N3,      HECCOM13
      3N4,     ODDC,   PABOVE, PBLO,   PI,     PIDTS,  PRINTL, PRINTS,  HECCOM14
      4PROB,   PRR,    PW,     RADE,   RC,     RFT,    RPTAG,  RR,      HECCOM15
      5SCDR,   SCR,    SCRE,   SCYCLE, SGNL,   SIG,    SIGH,   SLUB,    HECCOM16
      6SN,     SPROB,  SVMAX,  SVS,    S1,     S2,     S3,     S4,      HECCOM17
      7SS,     T,      TAUDTS, THICK,  TMAX,   TRAD,   URR,    UT,      HECCOM18
      8UU,     UVMAX,  VABOVE, VAPE,   VBLO,   VEL,    WS,     WSA,     HECCOM19
      9XMAXX,          BLANK(49)                                        HECCOM20
       COMMON          AIX(1200),      AMX(1200),      ALAMH(1200),
      1ALAMV(1200),    B(1200),        CAP(1200),      ER(1200),
      2FIOUT(1200),    KFIT(1200),     P(1200),        SMLQ(1200),
      3THETA(1200),    U(1200),        V(1200),        XALP(1200),
      4XB(1200),       SOLID(400),     PL(200),        Y(101),
      5DY(100),        GAMC(100),      TAMC(100),      UL(100),         HECCOM26
      6W2(100),        X(53),          DX(52),         TAU(52),         HECCOM27
      7BETAB(50),      TEMP(12),       HEAD(12)                         HECCOM28
C                                                                       DRA8038
C     THE FOLLOWING VARIABLES IN HECTIC COMMON HAVE BEEN RENAMED IN THIS
C     ROUTINE  --  CV TO CCVV , ISEND TO ISSSS , L TO LLLL , M TO MMMM,
C     AND XMAX TO XMAXX
C      ******************************************************************DRA8039
C                                                                       DRA8040
       DIMENSION       FLEFT(100),     PR(100),        SIGC(100)        DRA8041
       EQUIVALENCE     (FLEFT,UL),     (PL,PR)
       EQUIVALENCE    (PL(101),SIGC)
       EQUIVALENCE (S20,BLANK(12)), (ZP126,BLANK(13)), (ZP136,BLANK(14)),
      1(KMX,BLANK(15)),(ST,BLANK(16)),(ANUMBR,BLANK(19)),(LOCZ,BLANK(22))
      2,(H,BLANK(25)),(NA,BLANK(26))
       EQUIVALENCE (BACC,BLANK(47)),(SACC,BLANK(48)),(TACC,BLANK(49))
C                                                                       DRA8045
       REAL  MERGE                                                      DRA8046
C                                                                       DRA8047
C                                                                       DRA8048
       COMMON     /RAD1/    IAID,     KMAX,     ICHECK,   ISEND,        RAD1
      1L2,        L2P1,     LMAX,     M2,       MUMAX,    ISENDD,       RAD2
      2ISWEEP,    LLOW,     LHI                                         RAD
       COMMON     /RAD2/    ANGL1,    ANGL2,    ASN,      CHKSUM,       RAD3
```

```
1DELTY,     FORPI,    HELP1,    HELP2,    LMDA,     PP,              RAD4
2PI2,       PI4,      SINP2,    TANN,     TOLER,    XMIN,            RAD5
3XMAX,      STEF                                                     RAD6
 COMMON     /RAD3/              DUM1(1),            AA(16),          RAD7
1           COSPHI(6),          CVIN(6),            CVOUT(6),        RAD8
2           CH(6),              MU(6),              PHI(6),          RAD9
3           DUM2(1),            SIMLP(6),           SMU(6),          RAD10
4           W(6),               CV(6)                                RAD11
 COMMON    /INDEX/   M,L,LP                                          RAD12
 COMMON     /DRUMM/   LCHECK,   KK2,      NOBUFF                     RAD13
 COMMON     /DIS/     UV(6),    XV(6),    TV(6),    DH(6),           RAD14
1           XH(6),    YH(6)                                          RAD15
 COMMON     /SUMCHK/            SUMB(50,6),         SUMO(50,6)       RAD16
 COMMON     /BNDT/    BTHETA(50),    RTHETA(50),    ATHETA(50)       RAD17
C
 REAL    MU , LMDA
C
 EQUIVALENCE    (FIOUT,OLDTH) , (U,ROSS) ,
.    (CAP(1),BUFF1(1),BUFF(1),NOI) , (P(1),PLANCK(1),BUFF2(1),NOO)
 DIMENSION   OLDTH(1) , ROSS(1) , BUFF1(1) , BUFF(1) , PLANCK(1) ,
.  BUFF2(1)
C
 EQUIVALENCE   (V(1),XA(1),EB(1))
 DIMENSION   XA(1)
 DIMENSION            EB(50,6) , EV(50,6) , ET(50,6) ,
.           ETAIO(6,6) , ETAIT(6,6) , ETABI(6,6) , ETABT(6,6),
.  ETAUO(6,6) , ETAOI(6,6) , ETAOT(6,6) , ETACO(6,6)
C
C
 REAL   LMDA2
C
C  IG=1 IMPLIES GAMMA(I-1) DESIRED
C  IG=2 IMPLIES GAMMA(I) DESIRED     (NOTE -- TO CALCULATE GAMMA(I),
C    GAMMA(I-1) MUST BE CALCULATED)
C
  LMDA = TANN * (DELTY-Z)
C GAMMA(I-1)
  IF ( XMAX .LT. (XMIN+LMDA)  .AND.  I.NE.1  )  GO TO 10
  GAMMO = 0.
  GO TO 30
10 LMDA2 = LMDA*LMDA
  HLP = XMAX*XMAX - XMIN*XMIN
  IF ( LMDA2 .GT. HLP )  GO TO 20
  GAMMO = ACOS( (HLP+LMDA2)/(2.*LMDA*XMAX) )
  GO TO 30
20 GAMMO = ASN
30 IF ( IG .EQ. 1 )  RETURN
C GAMMA(I)
  HELP1 = GAMMO
  IF ( LMDA .GT. 2.*XMAX )  RETURN
  GAMMO = AMAX1( GAMMO, ACOS( LMDA/(2.*XMAX) ) )
  RETURN
  END
```

Subroutine TRAN2

```
C     TRAN2                                                             TRN1000
      SUBROUTINE TRAN2                                                  TRN1001
C    COMPILED 28 APRIL 1967, PHW
C    EB(I,L) = THE RATE AT WHICH ENERGY IS TRANSFERRED ACROSS FACE Y(J-1)
C      AT CELL X(I) IN FAN L                    (ERGS/SEC)
C    ET(I,L) = THE RATE AT WHICH ENERGY IS TRANSFERRED ACROSS FACE Y(J)
C      AT CELL X(I) IN FAN L                 (ERGS/SEC)
C    EV(IMAX,L) = THE RATE AT WHICH ENERGY IS TRANSFERRED ACROSS FACE
C      X(IMAX) IN FAN L FOR A PARTICULAR Y(J)          (ERGS/SEC)
C    EU(K) = THE NET RATE OF ENERGY DEPOSITION FOR CELL K  (ERGS/SEC)
C
C         H E C T I C   C O M M O N                                     TRN1003
C     **************************************************************** TRN1004
C                                                                       TRN1005
                                                                        TRN1006
      COMMON  VHN,     AMBIEN, ANN,    ATOM,   BCATAG, BCBTAG, BCLTAG,
     1BCHTA ,  CAPIN,  CAPS,   CUUT,   CMXK,   CMXOM,  CNDE,   CO,
     2CUL,     CSTOP,  CCVV,   CYCLE,  DBGPRT, DGREY,  DHNU,   DIANTP,   HECCOM03
     3DMIN,    DNN,    DT,     DTBUGR, DTC,    DTH,    DTNA,   DTQ,
     4DTH,     DTH1,   DTH2,   DTUF,   DTVF,   DUMPTT, ECK,    EFRAC,
     5EII,     ERRCNT, ETH,    EZERO,  FDTC,   FFA,    FFB,    FIFT,
     6FUDGE,   GV,     HCB,    HCP,    HH,     HNU,    HNUP,   HNUS,
     7HVW,     I,      IC,     IGOTO,  IH,     IHNU,   IMAX,   IMPTAG,
     8ISSSS,   ISR,    ITAG,   ITRMAX, IU,     IV,     I1,     I2,       HECCOM09
     9J,       JC,     JH,     JMAX,   JMAXA,  JU,     JV,     J5
      COMMON   K,      KMAXA,  KMN1,   KMN2,   KP,     LLLL,   MMMM,     HECCOM11
     1MERGE,   MFTAG,  MZ,     N,      NC,     NHNU,   NK,     NPC,
     2NR,      NRM,    NT,     NTAPE,  NY,     N1,     N2,     N3,
     3N4,      ODUC,   PABOVE, PBLO,   PI,     PIDTS,  PRINTL, PRINTS,
     4PROB,    PHH,    PW,     RADE,   RC,     RFT,    RPTAG,  RR,
     5SCDR,    SCR,    SCRE,   SCYCLE, SGNL,   SI3,    SIGH,   SLUG,
     6SN,      SPROB,  SVMAX,  SVS,    S1,     S2,     S3,     S4,
     7SS,      T,      TAUDTS, THICK,  TMAX,   TRAD,   URR,    UT,
     8UU,      UVMAX,  VABOVE, VAPE,   VBLO,   VEL,    WS,     WSA,
     9XMAXX,           BLANK(49)
      COMMON           AIX(1200),      AMK(1200),      ALAMH(1200),      HECCOM20
     1ALAMV(1200),     B(1200),        CAP(1200),      ER(1200),
     2FIOUT(1200),     KFIT(1200),     P(1200),        SMLQ(1200),
     3THETA(1200),     U(1200),        V(1200),        XALP(1200),
     4XB(1200),        SOLID(400),     PL(200),        Y(101),
     5DY(100),         GAML(100),      TAMC(100),      UL(100),          HECCOM26
     6W2(100),         X(53),          DX(52),         TAU(52),          HECCOM27
     7BETAB(50),       TEMP(12),       HEAD(12)                          HECCOM28
C                                                                       TRN1034
C    THE FOLLOWING VARIABLES IN HECTIC COMMON HAVE BEEN RENAMED IN THIS
C    ROUTINE  --  CV TO CCVV , ISEND TO ISSSS , L TO LLLL , M TO MMMM,
C    AND XMAX TO XMAXX
C     **************************************************************** TRN1035
C                                                                       TRN1036
      DIMENSION       FLEFT(100),     PR(100),        SIGC(100)          TRN1037
      EQUIVALENCE   (FLEFT,UL)  ,  (PL,PR)
      EQUIVALENCE    (PL(101),SIGC)
      EQUIVALENCE (S20,BLANK(12)), (ZP126,BLANK(13)), (ZP136,BLANK(14)),
     1(KMX,BLANK(15)),(WT,BLANK(16)),(ANUMBR,BLANK(19)),(LOCZ,BLANK(22))
     2,(H,BLANK(25)),(NA,BLANK(26))
      EQUIVALENCE  (BACC,BLANK(47)),(SACC,BLANK(48)),(TACC,BLANK(49))
```

```
C                                                                      TRN1041
      REAL  MERGE                                                      TRN1042
C                                                                      TRN1043
C                                                                      TRN1044
      COMMON   /RAD1/   IAID,    KMAX,    ICHECK,  ISEND,              RAD1
     1L2,      L2P1,    LMAX,    M2,      MUMAX,   ISENDC,             RAD2
     2ISWELP,  LLOW,    LHI                                            RAD
      COMMON   /RAD2/   ANGL1,   ANGL2,   ASN,     CHKSUM,             RAD3
     1DELTY,   FORPI,   HELP1,   HELP2,   LMDA,    PP,                 RAD4
     2PI2,     PI4,     SINP2,   TANN,    TOLER,   XMIN,               RAD5
     3XMAX,    STEF                                                    RAD6
      COMMON   /RAD3/            DUM1(1),          AA(16),             RAD7
     1         COSPHI(6),        CVIN(6),          CVOUT(6),           RAD8
     2         CH(6),            MU(6),            PHI(6),             RAD9
     3         DUM2(1),          SINLP(6),         SMU(6),             RAD10
     4         W(6),             CV(6)                                 RAD11
      COMMON   /INDEX/  M,L,LP                                         RAD12
      COMMON   /DRUMM/  LCHECK,  KK2,     NOBUFF                       RAD13
      COMMON   /DIS/    DV(6),   XV(6),   YV(6),   DH(6),              RAD14
     1         XH(6),   YH(6)                                          RAD15
      COMMON   /SUMCHK/          SUMB(50,6),       SUMO(50,6)          RAD16
      COMMON   /BNDT/   BTHETA(50),       RTHETA(50),   ATHETA(50)     RAD17
C
      REAL    MU , LMDA
C
      EQUIVALENCE   (FIOUT,OLDTH) , (U,ROSS) ,
     .   (CAP(1),BUFF1(1),BUFF(1),NOI) , (P(1),PLANCK(1),BUFF2(1),NOO)
      DIMENSION   OLDTH(1) , ROSS(1) , BUFF1(1) , BUFF(1) , PLANCK(1) ,
     .  BUFF2(1)
C
      EQUIVALENCE   (V(1),XA(1),EB(1))
      DIMENSION   XA(1)
      DIMENSION             EB(50,6) , EV(50,6) , ET(50,6) ,.
     .          ETAIO(6,6) , ETAIT(6,6) , ETABI(6,6) , ETABT(6,6),
     .  ETABO(6,6) , ETAOI(6,6) , ETAOT(6,6) , ETAOO(6,6)
C
      DIMENSION   PUR(50) , PUZ(100) , RUR(50) , RUZ(100)
C
C
      FOO = 0.0
      ETOP = 0.0
      ESIDE = 0.0
      EBTM = 0.0
      IF (NFTAG.EQ.0) DHNU=1.                                          TRN1059
      NVEZ=1                                                           TRN1060
      NY=NVEZ                                                          TRN1061
      IF (ITAG.EQ.0) NVEZ=2                                            TRN1062
      VEZ=NVEZ                                                         TRN1063
      CALL DVCHK (KDMY)                                                TRN1064
C    CALCULATE GEOMETRY FACTORS                                        TRN1067
      DO 10 I=1,IMAX                                                   TRN1068
      PUR(I)=P1/TAU(I)                                                 TRN1069
   10 RUR(I+1)=4.0E10*X(I)/(X(I+1)-X(I-1))                             TRN1070
      DO 20 J=1,JMAX                                                   TRN1071
      PUZ(J)=1./DY(J)
```

```
   20 RUZ(J+1)=2.0E10/(Y(J+1)-Y(J-1))                                   TRN1072
C   FIND HIGHEST TEMPERATURE
      THTAMX = .025
      DO 30 K=2,KMAX                                                    TRN1073
      IF (JMR(KFIT(K),2).NE.1) GO TO 30                                 TRN1074
      IF (THETA(K).LT.THTAMX) GO TO 30                                  TRN1075
      THTAMX=THETA(K)                                                   TRN1076
   30 CONTINUE                                                          TRN1077
      THTAMX=AMAX1(THTAMX,BCLTAG,BCRTAG,BCBTAG,BCATAG)                  TRN1078
C     REENTRY POINT FOR SECOND TEMPERATURE ITERATION                    TRN1079
   40 DO 50 K=2,KMAX                                                    TRN1080
      XA(K) = 0.
      XB(K) = 0.
   50 ER(K)=0.0                                                         TRN1083
C                                                                       TRN1084
C                                                                       TRN1085
C     ******************************************************************TRN1086
C                                                                      *TRN1087
C               D O   F R E Q U E N C Y   B O O K K E E P I N G        *TRN1088
C                                                                      *TRN1089
C     ******************************************************************TRN1090
C                                                                       TRN1091
C               SET UP MAX FREQ BOUNDARY                                TRN1092
C                                                                       TRN1093
      HNUP=1.0E6                                                        TRN1094
      HNUP4=1.0E24                                                      TRN1095
      IF (MFTAG.EQ.0) GO TO 180                                         TRN1096
   60 IHNU=IHNU+1                                                       TRN1097
      CALL KAPPA                                                        TRN1098
      HNU4=HNU**4                                                       TRN1099
      DHNU=HNUP-HNU                                                     TRN1100
C                                                                       TRN1101
C     MERGE GROUPS WITH HNU MORE THAN (MERGE) TIMES LARGEST THETA       TRN1102
C                                                                       TRN1103
      IF (MERGE.GT.0.0) GO TO 70                                        TRN1104
      S1=7.0225                                                         TRN1105
      GO TO 965
   70 IF (THTAMX-HNU/MERGE) 100,80,80                                   TRN1107
   80 IF (IHNU-1) 90,210,120                                            TRN1108
   90 S1=7.0235                                                         TRN1109
      GO TO 965
C                                                                       TRN1111
C               REJECT TAPE IF MORE THAN HALF OF GROUPS MERGE           TRN1112
C                                                                       TRN1113
  100 IF (IHNU+IHNU-NHNU) 120,110,110                                   TRN1114
  110 Q=AMOD(MERGE,1.)                                                  TRN1115
      IF (Q.GT.0.4.AND.Q.LT.0.6) GO TO 120                              TRN1116
      S1=7.0250                                                         TRN1117
      GO TO 965
  120 DO 130 K=2,KMAX                                                   TRN1119
      IF (JMR(KFIT(K),2).NE.1) GO TO 130                                TRN1120
      T4=THETA(K)**4                                                    TRN1121
      BETA=HNU/THETA(K)                                                 TRN1122
      BETAP=HNUP/THETA(K)                                               TRN1123
      DFU=PLNKUT(BETA,BETAP)                                            TRN1124
```

```
      IF (DFB.LT.1.E-10) GO TO 130                                      TRN1125
      TEMP(1)=DFB*T4                                                    TRN1126
      EMB1=EXP(-BETA)                                                   TRN1127
      EMB2=EXP(-BETAP)                                                  TRN1128
      TEMP(2)=DFB+0.0384974/T4*(HNU4/(1.0-EMB1)*EMB1-HNUP4/(1.0-EMB2)*EMTRN1129
     1B2)                                                               TRN1130
C                                                                       TRN1131
C              FORM NUMERATORS AND DENOMINATORS OF MERGED KAPPAS        TRN1132
C                                                                       TRN1133
      XA(K) = XA(K) + TEMP(1)
      XB(K) = XB(K) + TEMP(2)
      B(K) = B(K) + PLANCK(K)*TEMP(1)
      ER(K)=ER(K)+TEMP(2)/ROSS(K)                                       TRN1137
  130 CONTINUE                                                          TRN1138
      HNUP=HNU                                                          TRN1139
      HNUP4=HNU4                                                        TRN1140
      IF (THTAMX-HNU/MERGE) 60,140,140                                  TRN1141
C                                                                       TRN1142
C              FORM MERGED KAPPAS                                       TRN1143
C                                                                       TRN1144
 140  DO 174 K=2,KMAX
      IF (XA(K))  159,174,160
 159  S1 = 7.0520
      GO TO 965
 160  ROSS(K) = XB(K)/ER(K)
      PLANCK(K) = B(K)/XA(K)
      ER(K)=0.                                                          TRN1151
 174  CONTINUE
      HNUP=1.0E6                                                        TRN1153
      HNUP4=1.0E24                                                      TRN1154
      DHNU=HNUP-HNU                                                     TRN1155
      GO TO 230                                                         TRN1156
C                                                                       TRN1157
C              MONOFREQUENCY CALCULATION                                TRN1158
C                                                                       TRN1159
  180 NHNU=1                                                            TRN1160
      IHNU=1                                                            TRN1161
      HNU=.001                                                          TRN1166
      GO TO 230                                                         TRN1167
C     *****************************************************************TRN1168
C                                                                      *TRN1169
C               B E G I N   F R E Q U E N C Y   L O O P                *TRN1170
C                                                                      *TRN1171
C     *****************************************************************TRN1172
 205  IHNU = IHNU + 1
      CALL KAPPA                                                        TRN1174
      DHNU=HNUP-HNU                                                     TRN1175
      HNU4=HNU**4                                                       TRN1176
 210  CONTINUE
 230  CONTINUE
  270 CALL DVCHK (KDMY)                                                 TRN1205
      GO TO (280,290), KDMY                                             TRN1206
  280 S1=7.0522                                                         TRN1207
      GO TO 965
C                                                                       TRN1209
```

```
C                 FORM ROSSELAND AND PLANCK OPTICAL DEPTHS             TRN1210
C                                                                      TRN1211
C      DOUBLE ON STORAGE FOR ABSORPTION COEFFICIENTS, MU, AND LAMBDA    TRN1212
 290   DO 304 I=1,IMAX
       W2(I)=0.                                                        TRN1214
       K=I+1                                                           TRN1215
       M=K-IMAX                                                        TRN1216
       DO 304 J=1,JMAX
       FACTOR=1.                                                       TRN1218
       IF (ABS(RPTAU).GT.1.E-20) FACTOR=ROSS(K)/PLANCK(K)              TRN1219
       ROSS(K)=AMAX1(ROSS(K)*AMX(K)/(TAU(I)*DY(J)),1.E-20)
       PLANCK(K)=AMAX1(PLANCK(K)*FACTOR*AMX(K)/(TAU(I)*DY(J)),1.E-20)
       M=M+IMAX                                                        TRN1224
       K=K+IMAX                                                        TRN1225
 304   CONTINUE
C      ****************************************************************TRN1227
C                                                                     *TRN1228
C      T W O - D I M E N S I O N A L   T R A N S P O R T   S E C T I O N*TRN1229
C                                                                     *TRN1230
C      ****************************************************************TRN1231
       REWIND 3
       N = NOBUFF
       NB1 = N
       NBA = N*2
       NSKP = N*(JMAX-1)
       NOI = 0
C   ZERO  ER
 200   DO 150 J=1,JMAX
       DO 150 I=1,IMAX
       K = I + 1 + IMAX*(J-1)
 150   ER(K) = 0.
C
C
C   S T A R T   S W E E P I N G
C
       DO 5679 I=1,IMAX
       DO 5679 L=1,LMAX
       EB(I,L) = 0.
       EV(I,L) = 0.
       ET(I,L) = 0.
 5679  CONTINUE
C
       DO 2000 MMM=1,M2
       M = MMM
       FAC = -1.
       J = 1
       JADD = 1
       CALL  BOTBND
       GO TO 170
 155   M = MUMAX +1 -M
       FAC = 1.
       J = JMAX
       JADD = -1
       CALL  TOPBND
C   I N W A R D   S W E E P
```

```
170  ISWEEP = 1
     TH2 = Y(J) - DY(J)/2.
     YB = Y(J)
     IF (M .GT. M2)  YB=Y(J-1)
     I = IMAX
     CALL OUTBND
300  K = I + 1 + IMAX*(J-1)
     IACT = JMR(KFIT(K),2)
     SP = 4. * ROSS(K) * STEF * THETA(K) * THETA(K)
    .                         * THETA(K) * THETA(K)
     XHH = X(I-1) + DX(I)/2.
     IF (NOI .LE. 0)  GO TO 2200
350  IF (I .EQ. 1)  GO TO 650
     DO 600 L=1,L2
C    ETABI
     SUMBB = 0.
     DO 400 LP=1,L2
400  SUMBB = SUMBB + EB(I,LP)*BUFF(IBUFF+LP)
     IBUFF = IBUFF + L2
C    ETAOI
     SUMOO = 0.
     DO 500 LP=1,L2
500  SUMOO = SUMOO + EV(I,LP)*BUFF(IBUFF+LP)
     IBUFF = IBUFF + L2
     SUM = SUMBB + SUMOO
     D = BUFF(IBUFF+1)
     XAA = BUFF(IBUFF+2)
     TA = TH2 + FAC*BUFF(IBUFF+3)
     C = BUFF(IBUFF+4)
     IBUFF = IBUFF + 4
     EV(I-1,L) = TRANS(ROSS(K),D,XAA,X(I-1),TA,TH2,C,SUM,IACT,SP)
600  CONTINUE
C
650  DO 900 L=1,L2
C    ETABT
     SUMBB = 0.
     DO 700 LP=1,L2
700  SUMBB = SUMBB + EB(I,LP)*BUFF(IBUFF+LP)
     IBUFF = IBUFF + L2
C    ETAOT
     SUMOO = 0.
     DO 800 LP=1,L2
800  SUMOO = SUMOO + EV(I,LP)*BUFF(IBUFF+LP)
     IBUFF = IBUFF + L2
     SUM = SUMOO + SUMBB
     D = BUFF(IBUFF+1)
     XAA = BUFF(IBUFF+2)
     TA = TH2 + FAC*BUFF(IBUFF+3)
     C = BUFF(IBUFF+4)
     IBUFF = IBUFF + 4
     ET(I,L) = TRANS(ROSS(K),D,XAA,XHH,TA,YB,C,SUM,IACT,SP)
900  CONTINUE
C
     I = I - 1
     NOI = NOI - 1
```

```
1000 IF (I .LT. 1)  GO TO 1050
     GO TO 300
1050 CONTINUE
C
C   O U T W A R D   S W E E P
C
     ISWEEP = 2
     DO 1850 I=1,IMAX
     K = I + 1 + IMAX*(J-1)
     IACT = JMR(KFIT(K),2)
     SP = 4. * ROSS(K) * STEF * THETA(K) * THETA(K)
    *                         * THETA(K) * THETA(K)
     XHH = X(I-1) + DX(1)/2.
     IF (NOI .LE. 0)  GO TO 2200
1085 DO 1400 L=L2P1,LMAX
C   ETAIO
     SUMI = 0.
     DO 1100 LP=L2P1,LMAX
1100 SUMI = SUMI + EV(I-1,LP)*BUFF(IBUFF+LP-L2)
     IBUFF = IBUFF + L2
C   ETABO
     SUMBB = 0.
     DO 1200 LP=1,LMAX
1200 SUMBB = SUMBB + EB(I,LP)*BUFF(IBUFF+LP)
     IBUFF = IBUFF + LMAX
C   ETAOO
     SUMOO = 0.
     DO 1300 LP=1,L2
1300 SUMOO = SUMOO + EV(I,LP)*BUFF(IBUFF+LP)
     IBUFF = IBUFF + L2
     SUM = SUMI + SUMOO + SUMBB
     D = BUFF(IBUFF+1)
     XAA = BUFF(IBUFF+2)
     YA = YH2 + FAC*BUFF(IBUFF+3)
     C = BUFF(IBUFF+4)
     IBUFF = IBUFF + 4
     EV(I,L) = TRANS(ROSS(K),D,XAA,X(I),YA,YH2,C,SUM,IACT,SP)
1400 CONTINUE
C
     DO 1800 L=L2P1,LMAX
C   ETAIT
     SUMI = 0.
     DO 1500 LP=L2P1,LMAX
1500 SUMI = SUMI + EV(I-1,LP)*BUFF(IBUFF+LP-L2)
     IBUFF = IBUFF + L2
C   ETABT
     SUMBB = 0.
     DO 1600 LP=1,LMAX
1600 SUMBB = SUMBB + EB(I,LP)*BUFF(IBUFF+LP)
     IBUFF = IBUFF + LMAX
C   ETAOT
     SUMOO = 0.
     DO 1700 LP=1,L2
1700 SUMOO = SUMOO + EV(I,LP)*BUFF(IBUFF+LP)
     IBUFF = IBUFF + L2
```

```
      SUM = SUMI + SUMBB + SUMOO
      D = BUFF(IBUFF+1)
      XAA = BUFF(IBUFF+2)
      YA = YH2 + FAC*BUFF(IBUFF+3)
      C = BUFF(IBUFF+4)
      IBUFF = IBUFF + 4
      ET(I,L) = TRANS(ROSS(K),D,XAA,XHH,YA,YB,SUM,IACT,SP)
 1800 CONTINUE
      NOI = NOI - 1
 1850 CONTINUE
C
C   WRITE OUT E
 1851 IF (DBGPRT .LT. 1.E-20)  GO TO 1870
      WRITE (6,9005)  M,J
      IF ((J.NE.1.OR.M.GT.M2) .AND. (J.NE.JMAX.OR.M.LE.M2)) GO TO 1854
      WRITE (6,9050)
      DO 1853 I=1,IMAX
 1853 WRITE (6,9002)  I,(EB(I,L),L=1,LMAX)
 1854 WRITE (6,9001)
      DO 1855 I=1,IMAX
 1855 WRITE (6,9002)  I,(EV(I,L),L=1,LMAX)
      WRITE (6,9003)
      DO 1860 I=1,IMAX
 1860 WRITE (6,9002)  I,(ET(I,L),L=1,LMAX)
 1870 CONTINUE
C
C   COMPUTE ENERGY DEPOSITION RATES FOR  M,J
      DO 1874 I=1,IMAX
      K = I +1 + IMAX*(J-1)
      DO 1874 L=1,LMAX
      IF (L .GT. L2)  GO TO 1872
      HELP = EV(I-1,L)
      IF (I .EQ. 1)  HELP=0.
      ER(K) = ER(K) - HELP + EV(I,L)
      GO TO 1873
 1872 HELP = EV(I-1,L)
      IF (I .EQ. 1)  HELP=0.
      ER(K) = ER(K) + HELP - EV(I,L)
 1873 ER(K) = ER(K) + EB(I,L) - ET(I,L)
 1874 CONTINUE
C   COMPUTE SYSTEM ENERGY GAIN THRU BOUNDARIES
      IF (M.GT.M2 .OR. J.NE.1)  GO TO 1905
      DO 1900 I=1,IMAX
      DO 1900 L=1,LMAX
 1900 EBTM = EBTM + EB(I,L)
 1905 IF (M.GT.M2 .OR. J.NE.JMAX)  GO TO 1915
      IF (BCATAG .LT. 0.)  GO TO 1915
      DO 1910 I=1,IMAX
      DO 1910 L=1,LMAX
      W2(I) = W2(I) + ET(I,L)/TAU(I)
 1910 ETOP = ETOP - ET(I,L)
 1915 IF (M.LE.M2 .OR. J.NE.JMAX)  GO TO 1925
      IF (BCATAG .LT. 0.)  GO TO 1925
      DO 1920 I=1,IMAX
      DO 1920 L=1,LMAX
```

```
      W2(I) = W2(I) - EB(I,L)/TAU(I)
 1920 ETOP = ETOP + EB(I,L)
 1925 IF(M.LE.M2 .OR. J.NE.1)  GO TO 1935
      DO 1930 I=1,IMAX
      DO 1930 L=1-LMAX
 1930 EBTM = EBTM - ET(I,L)
 1935 DO 1945 L=1,LMAX
      IF (L .GT. L2)  GO TO 1940
      ESIDE = ESIDE + EV(IMAX,L)
      GO TO 1945
 1940 ESIDE = ESIDE - EV(IMAX,L)
 1945 CONTINUE
      DO 1875 I=1,IMAX
      DO 1875 L=1,LMAX
 1875 EB(I,L) = ET(I,L)
C
      J = J + JADD
      IF (J.LE.JMAX .AND. M.LE.M2)  GO TO 170
      IF (J .LE. JMAX)  GO TO 1890
      DO 1876 IJKK=1,NB1
 1876 BACKSPACE  3
      GO TO 155
 1890 IF (J .LT. 1)  GO TO 1895
      DO 1877 IJKK=1,NBA
 1877 BACKSPACE  3
      GO TO 170
 1895 IF (M .EQ. M2+1)  GO TO 2000
      DO 1878 IJKK=1,NSKP
 1878 READ(3)  (BUFF(ILM),ILM=1,1023)
 2000 CONTINUE
C
C
C   TAKE INTO ACCOUNT ALL DIRECTIONS
      DO 2005 K=2,KMAX
 2005 ER(K) = ER(K)*2.
      WRITE (6,9006)
      DO 2010 J=1,JMAX
      K1 = 2 + IMAX*(J-1)
      K2 = IMAX*J +1
      WRITE (6,9007)  J,(ER(K),K=K1,K2)
 2010 CONTINUE
      EBTM = EBTM * 2.
      ETOP = ETOP * 2.
      ESIDE = ESIDE * 2.
      FOO = EBTM + ESIDE + ETOP
      REWIND  3
      GO TO 835
C
C
 2200 READ (3)  (BUFF(ILM),ILM=1,1023)
      IF (EOF,3)  2211,2212
 2211 IF (ISWEEP.EQ.2 .AND. M.EQ.M2 .AND. J.EQ.JMAX .AND. I.EQ.IMAX)
     .  GO TO 2212
      IF (ISWEEP.EQ.2 .AND. M.EQ.M2+1 .AND. J.EQ.JMAX .AND. I.EQ.IMAX)
     .  GO TO 2212
```

```
      STOP
 2212 IBUFF = 1
      GO TO (350,1085) , ISWEEP
C
C
 835  HNUP = HNU
      HNUP=HNU                                                        TRN1381
      HNUP4=HNU4                                                      TRN1382
      IF (IHNU-NHNU)  205,850,840
 840  S1=7.1010                                                       TRN1384
      GO TO 965
 850  CONTINUE
 870  WRITE (6,970) NC                                                TRN1390
      IF (ABS(UGGPRT) .LT. 1.E-20) GO TO 880
      WRITE (6,980) (ER(K),K=2,KMAX)                                  TRN1391
C                                                                     TRN1392
C     ADVANCE FREQ, STORE EMERGENT FLUX, TEST FOR COMPLETION OF GROUPS TRN1393
C                                                                     TRN1394
C     ****************************************************************TRN1395
C                                                                    *TRN1396
C                E N D   F R E Q U E N C Y   L O O P                 *TRN1397
C                                                                    *TRN1398
C     ****************************************************************TRN1399
C     ITERATE ON TEMPERATURE                                          TRN1400
 880  GO TO (890,910), NVEZ                                           TRN1401
 890  NVEZ=2                                                          TRN1402
      VEZ=NVEZ                                                        TRN1403
      NT=NVEZ                                                         TRN1404
      I-HNU=0                                                         TRN1405
      DO 905 K=2,KMAX
C     WORK, SOURCE TERMS OMITTED                                      TRN1407
      IF (JMR(KFIT(K),2).NE.1)   GO TO 905
      OLDTH(K)=THETA(K)                                               TRN1409
      E=AIX(K)+ER(K)*DT/AMX(K)                                        TRN1410
      SV=TAU(I)+DT(J)/AMX(K)
      CALL LS (SV,E,TEMP(1),TEMP(2),GG)                               TRN1412
      THETA(K)=0.5*(THETA(K)+TEMP(1))                                 TRN1413
 905  CONTINUE
      IF (MFTAG.EQ.0) CALL KAPPA                                      TRN1415
      GO TO 40                                                        TRN1416
 910  IF (ITAG.EQ.0) GO TO 930                                        TRN1417
      DO 920 K=2,KMAX                                                 TRN1418
 920  THETA(K)=OLDTH(K)                                               TRN1419
C     R A N G E   I N T E R N A L   E N E R G I E S                   TRN1420
 930  TEMP(1)=1.                                                      TRN1421
      DO 940 K=2,KMAX                                                 TRN1422
      IF (JMR(KFIT(K),2).NE.1) GO TO 940                              TRN1423
      DE=ER(K)*DT/AMX(K)                                              TRN1424
      Q=SLUG*AIX(K)                                                   TRN1425
      IF(ABS(DE).LT.ABS(Q)) GO TO 940
      TEMP(1)=AMIN1(Q/ABS(DE),TEMP(1))                                TRN1427
 940  CONTINUE                                                        TRN1428
      DTEMP = DT * TEMP(1)
      IF (DTEMP .GT. FFB)  GO TO 950
      WRITE (6,970) NC                                                TRN1431
```

```
      WRITE (6,980) (ER(K),K=2,KMAX)                                   TRN1432
      WRITE (6,990) DE,K,AIX(K),AMX(K)                                 TRN1433
      S1=7.1066                                                        TRN1434
      GO TO 965
  950 T = T - DT
      DT = DTEMP
      T = T + DTEMP
      FOO = FOO * DT
      ETH = ETH + FOO
      BACC = BACC + EBTM*DT
      SACC = SACC + ESIDE*DT
      TACC = TACC + ETOP*DT
C     FOO,ETH,Z ARE IN ERGS
      WRITE (6,1001)  NC,DT
      WRITE (6,1010)  BACC,SACC,TACC,FOO
      READ (4)  FIOUT,CAP,P,U,V
      REWIND  4
      RETURN                                                           TRN1451
  967 S1 = 7.1111
  965 READ (4)  FIOUT,CAP,P,U,V
      ISEND = 2
      CALL EDIT
C                                                                      TRN1452
  970 FORMAT (52H0INTERNAL ENERGY GAIN PER UNIT TIME DUE TO RADIATION,8H
     1 CYCLE =15)                                                      TRN1454
  980 FORMAT (1X,1P10E11.4)                                            TRN1455
  990 FORMAT (5H DE =1PE11.4,5H  K =I10,6H AIX =E11.4,6H AMX =E11.4)   TRN1456
 1001 FORMAT (19H0   FLUX FOR CYCLE I4,7H   DT =1PE10.3)
 1010 FORMAT (9H BOTTOM =1PE11.4,6H  SIDE =E11.4,7H  TOP =E11.4,7H  FOO TRN1458
     1=E11.4)                                                          TRN1459
 9000 FORMAT (7H1FOR M=,I2,3H J=,I2,24H ER FOR INWARD SWEEP ARE)
 9001 FORMAT (1H0//10X,7HEV(I,L)/50X,1HL/)
 9002 FORMAT (3H I=,I2/(6X,1P8E14.6))
 9003 FORMAT (1H0//10X,7HET(I,L)/50X,1HL/)
 9005 FORMAT (7H1FOR M=,I2,3H J=,I2,17H ENERGY RATES ARE)
 9006 FORMAT (28H1NET ENERGY DEPOSITION RATES//50X,1HI/)
 9007 FORMAT (3H J=,I2/(6X,1P8E14.6))
 9012 FORMAT (9H1 BOTTOM=,1PE14.6,5X,5HSIDE=,1PE14.6,5X,4HTOP=,1PE14.6,5
     ,X,4HFOO=,1PE14.6)
 9050 FORMAT (1H0//10X,7HEB(I,L)/50X,1HL/)
 9010 FORMAT (1H0/(1X,1P10E11.4))
      END                                                              TRN1460
```

Function TRANS(SIGMA, D, XA, XB, YA, YB, C, SUM, IACT, SP)

```
      FUNCTION  TRANS(SIGMA,D,XA,XB,YA,YB,C,SUM,IACT,SP)
C     TRANSPORT FROM (XA,YA) TO (XB,YB)
C     FORPI = 12.566371
      DATA   FAC/2./
      IF (IACT .NE. 1)  GO TO 100
      A1 = SIGMA * D
      A2 = 12.566371 * SIGMA
      A3 = A2 * SIGMA
      A4 = EXP(-A1)
      A5 = 1. - A4
      SA = S(XA,YA)
      SB = S(XB,YB)
      HELP = AMIN1(SA,SB,SP)
      IF ( ( AMAX1(SA,SB,SP)-HELP)/HELP .LT. FAC )  GO TO 90
      SA = SP
      SB = SP
 90   TRANS = SUM*A4 + C*( SA/A2*A5 +(SB-SA)/A3*(A1-A5)/D)
      RETURN
C     TRANSFER THRU AN INACTIVE ZONE
 100  TRANS = SUM
      RETURN
      END
```

Function S(XX, YY)

```
      FUNCTION   S(XX,YY)
C   S(XX,YY)  =
C             THE SOURCE (RATE AT WHICH ENERGY IS BEING GENERATED PER
C             UNIT VOLUME   --  ERGS/(CM**3 SEC) )  AT POINT (XX,YY)
C
C    ****************************************************************DRA8007
C        H E C T I C   C O M M O N                                   DRA8008
C    ****************************************************************DRA8009
      COMMON  AHN,     AMBIEN,  ANN,     ATOM,    BCATAG,  BCBTAG,  BCLTAG,  HECCOM01
     1BCRTAG, CAPIN,   CAPS,    CDUT,    CMXK,    CMXOM,   CNDE,    CO,      HECCOM02
     2COE,    CSTOP,   CCVV,    CYCLE,   DBGPRT,  DGREY,   DHNU,    DIANTP,  HECCOM03
     3DMIN,   DMN,     DT,      DTBUGR,  DTC,     DTH,     DTNA,    DTQ,     HECCOM04
     4DTR,    DTR1,    DTR2,    DTUF,    DTVF,    DUMPT7,  ECK,     EFRAC,   HECCOM05
     5EI1,    ERRCRT,  ETH,     EZERO,   FDIC,    FFA,     FFB,     FIFT,    HECCOM06
     6FUDGE,  GV,      HCB,     HCP,     HH,      HNU,     HNUP,    HNUS,    HECCOM07
     7HVB,    I,       IC,      IGOTO,   IH,      IHNU,    IMAX,    IMPTAG,  HECCOM08
     8ISSSS,  ISR,     ITAG,    ITRMAX,  IU,      IV,      I1,      I2,      HECCOM09
     9J,      JC,      JH,      JMAX,    JMAXA,   JU,      JV,      J5       HECCOM10
      COMMON  K,       KMAXA,   KMN1,    KMN2,    KP,      LLLL,    MMMM,    HECCOM11
     1MERGE,  MFTAG,   M2,      N,       NC,      NHNU,    NK,      NPC,     HECCOM12
     2NR,     NRM,     NT,      NTAPE,   NY,      N1,      N2,      N3,      HECCOM13
     3N4,     ODDC,    PABOVE,  PBLO,    P1,      PIDTS,   PRINTL,  PRINTS,  HECCOM14
     4PROB,   PRR,     PW,      RADE,    RC,      RFT,     RPTAG,   RR,      HECCOM15
     5SCUR,   SCR,     SCRE,    SCYCLE,  SGNL,    SIG,     SIGH,    SLUG,    HECCOM16
     6SN,     SPROB,   SVMAX,   SVS,     S1,      S2,      S3,      S4,      HECCOM17
     7SS,     T,       TAUDTS,  THICK,   TMAX,    TRAD,    URR,     UT,      HECCOM18
     8UU,     UVMAX,   VABOVE,  VAPE,    VULO,    VEL,     WS,      WSA,     HECCOM19
     9XMAXX,           BLANK(49)                                             HECCOM20
      COMMON           AIX(1200),        AMX(1200),        ALAMH(1200),
     1ALAMV(1200),     B(1200),          CAP(1200),        ER(1200),
     2FIOUT(1200),     KFIT(1200),       P(1200),          SMLO(1200),
     3THETA(1200),     U(1200),          V(1200),          XALP(1200),
     4XB(1200),        SOLID(400),       PL(200),          Y(101),
     5DY(100),         GAMC(100),        YAMC(100),        UL(100),        HECCOM26
     6W2(100),         X(53),            UX(52),           TAU(52),        HECCOM27
     7DETAB(50),       TEMP(12),         HEAU(12)                          HECCOM28
C                                                                        DRA8038
C    THE FOLLOWING VARIABLES IN HECTIC COMMON HAVE BEEN RENAMED IN THIS
C    ROUTINE  --  CV TO CCVV , ISEND TO ISSSS , L TO LLLL , M TO MMMM,
C    AND XMAX TO XMAXX
C      **************************************************************DRA8039
C                                                                        DRA8040
       DIMENSION       FLEFT(100),      PR(100),        SIGC(100)        DRA8041
       EQUIVALENCE     (FLEFT,UL),      (PL,PR)
       EQUIVALENCE     (PL(101),SIGC)
       EQUIVALENCE (S20,BLANK(12)), (ZP126,BLANK(13)), (ZP136,BLANK(14)),
      1(KMX,BLANK(15)),(WT,BLANK(16)),(ANUMBR,BLANK(19)),(LOCZ,BLANK(22))
      2,(H,BLANK(25)),(NA,BLANK(26))
       EQUIVALENCE (BACC,BLANK(47)),(SACC,BLANK(48)),(TACC,BLANK(49))
C                                                                        DRA8045
       REAL   MERGE                                                      DRA8046
C                                                                        DRA8047
C                                                                        DRA8048
C    GIVEN CELL TEMPERATURES , COMPUTE CELL-CENTERED SOURCE TERMS TO
C    OBTAIN THE SOURCE AT  (XX,YY) WITH BILINEAR INTERPOLATION
C
```

```
C   RELATIONSHIPS --
C      (STEFANS CONST.)   =  AC/4   =   1.0283E+12
C                                        ENGS/(CM**2 SEC EV**4)
C     SOURCE   =  S  =   4.*ROSS*(STEFANS CONST.)*(THETA**4)
C                                        ENGS/(CM**3 SEC)
C     INTENSITY OF RADIATION EMITTED BY A BLACK SURFACE OF TEMP,
C        THETA INTEGRATED OVER ALL FREQUENCIES
C            =  B  =   S/(4.*PI*ROSS)
C                  =   (STEFANS CONST.)*(THETA**4)/PI
C                                        ENGS/(CM**2 SEC STERADIAN)
      COMMON    /RAD1/    IAID,     KMAX,     ICHECK,   ISEND,          RAD1
     1L2,       L2P1,     LMAX,     M2,       MUMAX,    ISENDD,         RAD2
     2ISWEEP,   LLOW,     LMI                                           RAD
      COMMON    /RAD2/    ANGL1,    ANGL2,    ASN,      CHKSUM,         RAD3
     1DELTT,    FORPI,    HELP1,    HELP2,    LMDA,     PP,             RAD4
     2PI2,      PI4,      SINP2,    TANN,     TOLER,    XMIN,           RAD5
     3XMAX,     STEF                                                    RAD6
      COMMON    /PAD3/              DUM1(1),            AA(16),         RAD7
     1          COSPHI(6),          CVIN(6),            CVOUT(6),       RAD8
     2          CH(6),              MU(6),              PHI(6),         RAD9
     3          DUM2(1),            SINLP(6),           SMU(6),         RAD10
     4          W(6),               CV(6)                               RAD11
      COMMON   /INDEX/   M,L,LP                                         RAD12
      COMMON    /DRUMM/   LCHECK,   KK2,      NOBUFF                    RAD13
      COMMON    /DIS/     DV(6),    XV(6),    TV(6),    DH(6),          RAD14
     1          XH(6),    YH(6)                                         RAD15
      COMMON    /SUMCHK/            SUMB(50,6),         SUMO(50,6)      RAD16
      COMMON    /BNDT/    BTHETA(50),       RTHETA(50),    ATHETA(50)   RAD17
C
      REAL    MU , LMDA
C
      EQUIVALENCE    (FIOUT,OLDTH) , (U,ROSS) ,
     .   (CAP(1),BUFF1(1),BUFF(1),NOI) , (P(1),PLANCK(1),BUFF2(1),NOO)
      DIMENSION   OLDTH(1) , ROSS(1) , BUFF1(1) , BUFF(1) , PLANCK(1) ,
     .   BUFF2(1)
C
      EQUIVALENCE    (V(1),XA(1),EB(1))
      DIMENSION   XA(1)
      DIMENSION             EB(50,6) , EV(50,6) , ET(50,6) ,
     .          ETAIO(6,6) , ETAIT(6,6) , ETABI(6,6) , ETABT(6,6),
     .   ETABO(6,6) , ETAOI(6,6) , ETAOT(6,6) , ETAOO(6,6)
C
C
C
C    THETA  IN  EV  FROM  KAPPA
       JLOW = J-1
       IF (TT .LT. T(J)-DT(J)/2.)   GO TO 100
       JLOW = J
 100   ILFT = I-1
       IF (XX .LT. X(I)-DX(I)/2.)   GO TO 200
       ILFT = I
 200   KLL = ILFT +1 +IMAX*(JLOW-1)
       KUL = ILFT +1 +IMAX*(JLOW)
       DXP = ( DX(ILFT+1) + DX(ILFT) )/2.
       DYP = ( DY(JLOW+1) + DT(JLOW) )/2.
```

```
      YUP = Y(JLOW+1) - DY(JLOW+1)/2.
      YDN = Y(JLOW) - DY(JLOW)/2.
      XRGT = X(ILFT+1) - DX(ILFT+1)/2.
      XLFT = X(ILFT) - DX(ILFT)/2.
      IF (JLOW.EQ.0  .AND.  ILFT.EQ.IMAX )  GO TO 700
      IF (JLOW.EQ.JMAX  .AND.  ILFT.EQ.IMAX)  GO TO 600
      IF (JLOW.EQ.JMAX  .AND.  ILFT.EQ.0)  GO TO 575
      IF (JLOW.EQ.0  .AND.  ILFT.EQ.0)  GO TO 550
      IF (ILFT .EQ. IMAX)  GO TO 500
      IF (ILFT  .EQ.  0)  GO TO 450
      IF (JLOW .EQ. JMAX)  GO TO 400
      IF (JLOW .EQ. 0)  GO TO 300
C   NORMAL  BILINEAR  INTERPOLATION
      DELT = DXP * DYP
      ALL = (YUP-YY)*(XRGT-XX)/DELT
      ALR = (YUP-YY)*(XX-XLFT)/DELT
      AUL = (YY-YDN)*(XRGT-XX)/DELT
      AUR = (YY-YDN)*(XX-XLFT)/DELT
      S = SRC(KLL)*ALL + SRC(KLL+1)*ALR + SRC(KUL)*AUL + SRC(KUL+1)*AUR
      RETURN
C   LOWER BOUNDARY
 300  S = SRC(KUL) + ( SRC(KUL+1)-SRC(KUL) )/DXP * (XX-XLFT)
      RETURN
C   UPPER BOUNDARY
 400  S = SRC(KLL) + ( SRC(KLL+1)-SRC(KLL) )/DXP * (XX-XLFT)
      RETURN
C   AXIS
 450  S = SRC(KLL+1) + ( SRC(KUL+1)-SRC(KLL+1) )/DYP * (YY-YDN)
      RETURN
C   SIDE BOUNDARY
 500  S = SRC(KLL) + ( SRC(KUL)-SRC(KLL) )/DYP * (YY-YDN)
      RETURN
C   LOWER INSIDE CORNER
 550  S = SRC(KUL+1)
      RETURN
C   UPPER INSIDE CORNER
 575  S = SRC(KLL+1)
      RETURN
C   UPPER OUTSIDE CORNER
 600  S = SRC(KLL)
      RETURN
C   LOWER OUTSIDE CORNER
 700  S = SRC(KUL)
      RETURN
      END
```

Function SRC(KKK)

```
      FUNCTION    SRC(KKK)
C     ******************************************************************DRA8007
C          H E C T I C     C O M M O N                                  DRA8008
C     ******************************************************************DRA8009
      COMMON  AHN,    AMBIEN, ANN,    ATOM,   BCATAG, BCBTAG, BCLTAG,   HECCOM01
     1BCRTAG, CAPIN,  CAPS,   COUT,   CMXK,   CMXOM,  CNDE,   CO,       HECCOM02
     2COE,    CSTOP,  CCVV,   CYCLE,  DBGPRT, DGREY,  DHNU,   DIANTP,   HECCOM03
     3DMIN,   DNN,    DT,     DTBUGR, DTC,    DTH,    DTNA,   DTO,      HECCOM04
     4DTR,    DTR1,   DTR2,   DTUF,   DTVF,   DUMPT7, ECK,    EFRAC,    HECCOM05
     5EI1,    ERRCRT, ETH,    EZERO,  FDTC,   FFA,    FFB,    FIFT,     HECCOM06
     6FUDGE,  GV,     HCB,    HCP,    HH,     HNU,    HNUP,   HNUS,     HECCOM07
     7HVB,    I,      IC,     IGOTO,  IH,     IHNU,   IMAX,   IMPTAG,   HECCOM08
     8ISSSS,  ISR,    ITAG,   ITRMAX, IU,     IV,     I1,     I2,       HECCOM09
     9J,      JC,     JP,     JMAX,   JMAXA,  JU,     JV,     J5        HECCOM10
      COMMON  K,      KMAXA,  KMN1,   KMN2,   KP,     LLLL,   MMMM,     HECCOM11
     1MERGE,  MFTAG,  MZ,     N,      NC,     NHNU,   NK,     NPC,      HECCOM12
     2NR,     NRM,    NT,     NTAPE,  NY,     N1,     N2,     N3,       HECCOM13
     3N4,     ODDC,   PABOVE, PBLO,   PI,     PIDTS,  PRINTL, PRINTS,   HECCOM14
     4PROB,   PRR,    PW,     RADE,   RC,     RFT,    RPTAG,  RR,       HECCOM15
     5SCDR,   SCR,    SCRE,   SCYCLE, SGNL,   SIG,    SIGH,   SLUG,     HECCOM16
     6SN,     SPROB,  SVMAX,  SVS,    S1,     S2,     S3,     S4,       HECCOM17
     7SS,     T,      TAUDTS, THICK,  TMAX,   TRAD,   URR,    UT,       HECCOM18
     8UU,     UVMAX,  VABOVE, VAPE,   VBLO,   VEL,    WS,     WSA,      HECCOM19
     9XMAXX,          BLANK(49)                                         HECCOM20
      COMMON          AIX(1200),      AMX(1200),      ALAMH(1200),
     1ALAMV(1200),    B(1200),        CAP(1200),      ER(1200),
     2FIOUT(1200),    KFIT(1200),     P(1200),        SMLQ(1200),
     3THETA(1200),    U(1200),        V(1200),        XALP(1200),
     4XB(1200),       SOLID(400),     PL(200),        Y(101),
     5DY(100),        GAMC(100),      YAMC(100),      UL(100),          HECCOM26
     6W2(100),        X(53),          DX(52),         TAU(52),          HECCOM27
     7BETAB(50),      TEMP(12),       HEAD(12)                          HECCOM28
C                                                                       DRA8038
C     THE FOLLOWING VARIABLES IN HECTIC COMMON HAVE BEEN RENAMED IN THIS
C     ROUTINE  --  CV TO CCVV , ISEND TO ISSSS , L TO LLLL , M TO MMMM,
C     AND XMAX TO XMAXX
C     ******************************************************************DRA8039
C                                                                       DRA8040
      DIMENSION       FLEFT(100),     PR(100),        SIGC(100)         DRA8041
      EQUIVALENCE     (FLEFT,UL),     (PL,PR)
      EQUIVALENCE     (PL(101),SIGC)
      EQUIVALENCE (S20,BLANK(12)), (ZP126,BLANK(13)), (ZP136,BLANK(14)),
     1(KMX,BLANK(15)),(WT,BLANK(16)),(ANUMBR,BLANK(19)),(LOCZ,BLANK(22))
     2,(H,BLANK(25)),(NA,BLANK(26))
      EQUIVALENCE (BACC,BLANK(47)),(SACC,BLANK(48)),(TACC,BLANK(49))
C                                                                       DRA8045
      REAL  MERGE                                                       DRA8046
C                                                                       DRA8047
C                                                                       DRA8048
      COMMON    /RAD1/    IAID,     KMAX,     ICHECK,   ISEND,          RAD1
     1L2,       L2P1,     LMAX,     M2,       MUMAX,    ISENDD,         RAD2
     2ISWEEP,   LLO,      LHI                                           RAD
      COMMON    /RAD2/    ANGL1,    ANGL2,    ASN,      CHKSUM,         RAD3
     1DELTY,    FORPI,    HELP1,    HELP2,    LMDA,     PP,             RAD4
     2PI2,      PI4,      SINP2,    TANN,     TOLER,    XMIN,           RAD5
     3XMAX,     STEF                                                    RAD6
```

```
      COMMON    /RAD3/                DUM1(1),          AA(16),         RAD7
     1          COSPHI(6),            CVIN(6),          CVOUT(6),       RAD8
     2          CH(6),                MU(6),            PHI(6),         RAD9
     3          DUM2(1),              SINLP(6),         SMU(6),         RAD10
     4          W(6),                 CV(6)                             RAD11
      COMMON   /INDEX/   M,L,LP                                         RAD12
      COMMON    /DRUMM/   LCHECK,   KK2,      NOBUFF                    RAD13
      COMMON    /DIS/     DV(6),    XV(6),    YV(6),    DH(6),          RAD14
     1          XH(6),    YH(6)                                         RAD15
      COMMON    /SUMCHK/              SUMB(50,6),       SUMO(50,6)      RAD16
      COMMON    /BNDT/    BTHETA(50),       RTHETA(50),    ATHETA(50)   RAD17
C
      REAL     MU , LMDA
C
      EQUIVALENCE   (FIOUT,OLDTH) , (U,ROSS) ,
     .   (CAP(1),BUFF1(1),BUFF(1),NOI) , (P(1),PLANCK(1),BUFF2(1),NOO)
      DIMENSION   OLDTH(1) , ROSS(1) , BUFF1(1) , BUFF(1) , PLANCK(1) ,
     .  BUFF2(1)
C
      EQUIVALENCE   (V(1),XA(1),EB(1))
      DIMENSION   XA(1)
      DIMENSION             EB(50,6) , EV(50,6) , ET(50,6) ,
     .            ETAIO(6,6) , ETAIT(6,6) , ETABI(6,6) , ETABT(6,6),
     .  ETABO(6,6) , ETAOI(6,6) , ETAOT(6,6) , ETAOO(6,6)
C
C
C
      IF (JMH(KFIT(KKK),2) .EQ. 1)  GO TO 50
C   INACTIVE CELL
      SRC = 0.
      RETURN
C   ACTIVE CELL
 50   SRC    = 4. * ROSS(K) * STEF * THETA(KKK) * THETA(KKK)
     .                             * THETA(KKK) * THETA(KKK)
      IF (MFTAG .EQ. 0)  GO TO 100
C   MULTIFREQUENCY CASE
      DFB = PLNKUT(HNU/THETA(KKK),HNUP/THETA(KKK))
      SRC = SRC * DFB
 100  RETURN
      END
```

Subroutine BOTBND

```
      SUBROUTINE  BOTBND
C   BOUNDARY  CONDITIONS  FOR  BOTTOM  SURFACE
C     ***************************************************************DRA8007
C         H E C T I C   C O M M O N                                  DRA8008
C     ***************************************************************DRA8009
      COMMON  AHN,    AMBIEN, ANN,    ATOM,   BCATAG, BCBTAG, BCLTAG,   HECCOM01
     1BCNTAG, CAPIN,  CAPS,   COUT,   CMXK,   CMXOM,  CNDE,   CO,       HECCOM02
     2COE,    CSTOP,  CCVV,   CYCLE,  DBGPRT, DGREY,  DHNU,   DIANTP,   HECCOM03
     3DMIN,   DNN,    DT,     DTBUGR, DTC,    DTH,    DTNA,   DTO,      HECCOM04
     4DTR,    DTR1,   DTR2,   DTUF,   DTVF,   DUMPT7, ECK,    EFRAC,    HECCOM05
     5EI1,    EHRCHT, ETH,    EZERO,  FDTC,   FFA,    FFB,    FIFT,     HECCOM06
     6FUDGE,  GV,     HCB,    HCP,    HH,     HNU,    HNUP,   HNUS,     HECCOM07
     7HVB,    I,      IC,     IGOTO,  IH,     IHNU,   IMAX,   IMPTAG,   HECCOM08
     8ISSSS,  ISR,    ITAG,   ITRMAX, IU,     IV,     I1,     I2,       HECCOM09
     9J,      JC,     JH,     JMAX,   JMAXA,  JU,     JV,     J5        HECCOM10
      COMMON  K,      KMAXA,  KMN1,   KMN2,   KP,     LLLL,   MMMM,     HECCOM11
     1MERGE,  MFTAG,  MZ,     N,      NC,     NHNU,   NK,     NPC,      HECCOM12
     2NR,     NRM,    NT,     NTAPE,  NY,     N1,     N2,     N3,       HECCOM13
     3N4,     ODDC,   PABOVE, PBLO,   PI,     PIDTS,  PRINTL, PRINTS,   HECCOM14
     4PROB,   PRR,    PW,     RADE,   RC,     RFT,    RPTAG,  RR,       HECCOM15
     5SCDR,   SCR,    SCHE,   SCYCLE, SGNL,   SIG,    SIGH,   SLUG,     HECCOM16
     6SN,     SPROB,  SVMAX,  SVS,    S1,     S2,     S3,     S4,       HECCOM17
     7S5,     T,      TAULTS, THICK,  TMAX,   TRAD,   URR,    UT,       HECCOM18
     8UU,     UVMAX,  VABOVE, VAPE,   VBLO,   VEL,    WS,     WSA,      HECCOM19
     9XMAXX,          BLANK(49)                                         HECCOM20
      COMMON          AIX(1200),     AMX(1200),     ALAMH(1200),
     1ALAMV(1200),    B(1200),       CAP(1200),     ER(1200),
     2FIOUT(1200),    KFIT(1200),    P(1200),       SMLO(1200),
     3THETA(1200),    U(1200),       V(1200),       XALP(1200),
     4XB(1200),       SOLID(400),    PL(200),       Y(101),
     5UT(100),        GAMC(100),     YAMC(100),     UL(100),          HECCOM26
     6W2(100),        X(53),         DX(52),        TAU(52),          HECCOM27
     7BETAB(50),      TEMP(12),      HEAD(12)                         HECCOM28
C                                                                       DRA8038
C   THE FOLLOWING VARIABLES IN HECTIC COMMON HAVE BEEN RENAMED IN THIS
C   ROUTINE  --  CV TO CCVV , ISEND TO ISSSS , L TO LLLL , M TO MMMM,
C   AND XMAX TO XMAXX
C     ***************************************************************DRA8039
C                                                                       DRA8040
      DIMENSION       FLEFT(100),    PR(100),       SIGC(100)           DRA8041
      EQUIVALENCE     (FLEFT,UL),    (PL,PR)
      EQUIVALENCE    (PL(101),SIGC)
      EQUIVALENCE  (S20,BLANK(12)), (ZP126,BLANK(13)), (ZP136,BLANK(14)),
     1(KMX,BLANK(15)),(WT,BLANK(16)),(ANUMBR,BLANK(19)),(LOCZ,BLANK(22))
     2,(M,BLANK(25)),(NA,BLANK(26))
      EQUIVALENCE  (BACC,BLANK(47)),(SACC,BLANK(48)),(TACC,BLANK(49))
C                                                                       DRA8045
      REAL  MERGE                                                       DRA8046
C                                                                       DRA8047
C                                                                       DRA8048
      COMMON   /RAD1/    IA1D,     KMAX,     ICHECK,   ISEND,           RAD1
     1L2,      L2P1,     LMAX,     M2,       MUMAX,    ISENDD,          RAD2
     2ISWEEP,  LLOW,     LHI                                            RAD
      COMMON   /RAD2/    ANGL1,    ANGL2,    ASN,      CHKSUM,          RAD3
     1DELTY,   FORPI,    HELP1,    HELP2,    LMDA,     PP,              RAD4
```

```
     2PI2,       PI4,        SIHP2,     TANH,      TOLER,     XMIN,            RAD5
     3XMAX,      STEF                                                          RAD6
      COMMON     /RAD3/                 DUM1(1),              AA(16),          RAD7
     1           COSPHI(6),             CVIN(6),              CVOUT(6),        RAD8
     2           CH(6),                 MU(6),                PHI(6),          RAD9
     3           DUM2(1),               SINLP(6),             SMU(6),          RAD10
     4           W(6),                  CV(6)                                  RAD11
      COMMON     /INDEX/   M,L,LP                                              RAD12
      COMMON     /DRUMM/   LCHECK,    KK2,       NOBUFF                        RAD13
      COMMON     /DIS/     DV(6),     XV(6),     YV(6),     DH(6),             RAD14
     1           XH(6),    YH(6)                                               RAD15
      COMMON     /SUMCHK/               SUMB(50,6),           SUMO(50,6)       RAD16
      COMMON     /BNDT/    BTHETA(50),        RTHETA(50),     ATHETA(50)       RAD17
C
      REAL    MU , LMDA
C
      EQUIVALENCE    (FIOUT,OLDTH) , (U,ROSS) ,
     .    (CAP(1),BUFF1(1),BUFF(1),NOI) , (P(1),PLANCK(1),BUFF2(1),NOO)
      DIMENSION   OLDTH(1) , ROSS(1) , BUFF1(1) , BUFF(1) , PLANCK(1) ,
     .  BUFF2(1)
C
      EQUIVALENCE    (V(1),XA(1),EB(1))
      DIMENSION   XA(1)
      DIMENSION              EB(50,6) , EV(50,6) , ET(50,6) ,
     .           ETAIO(6,6) , ETAIT(6,6) , ETABI(6,6) , ETABT(6,6),
     .  ETABO(6,6) , ETAOI(6,6) , ETAOT(6,6) , ETAOO(6,6)
C
C
      REAL   JB
C
C   EB(I,L) = THE RATE AT WHICH ENERGY IS TRANSFERRED ACROSS FACE Y(J-1)
C     AT CELL X(I) IN FAN L              (ERGS/SEC)
      DO 1000 I=1,IMAX
      DO 900 L=0,LMAX
  900 AA(L) = SINLP(L) * X(I)
      DO 1000 L=1,LMAX
      BINT = STEF/PI * BTHETA(I) * BTHETA(I)
     .               * BTHETA(I) * BTHETA(I)
 1000 EB(I,L) = BINT * JB(J-1,I,L,M)
      RETURN
      END
```

Subroutine OUTBND

```
      SUBROUTINE    OUTBND
C     BOUNDARY  CONDITIONS  FOR  CURVED  OUTSIDE  SURFACE
C     ****************************************************************DRA8007
C           H E C T I C   C O M M O N                                  DRA8008
C     ****************************************************************DRA8009
      COMMON   AHN,     AMBIEN,  ANN,     ATOM,    BCATAG,  BCHTAG,  BCLTAG,   HECCOM01
     1BCRTAG,  CAPIN,   CAPS,    COUT,    CMXK,    CMXOM,   CNDE,    CO,       HECCOM02
     2COE,     CSTOP,   CCVV,    CYCLE,   DBGPRT,  DGHEY,   DHNU,    DIANTP,   HECCOM03
     3DMIN,    DNN,     DT,      DTBUGR,  DTC,     DTH,     DTNA,    DTO,      HECCOM04
     4DTR,     DTR1,    DTR2,    DTUF,    DTVF,    DUMPT7,  ECK,     EFRAC,    HECCOM05
     5E11,     ERRCRT,  ETH,     EZERO,   FDTC,    FFA,     FFB,     F1FT,     HECCOM06
     6FUDGE,   GV,      HCB,     HCP,     HH,      HNU,     HNUP,    HNUS,     HECCOM07
     7HVB,     I,       IC,      IGOTO,   IH,      IHNU,    IMAX,    IMPTAG,   HECCOM08
     81SSSS,   ISR,     ITAG,    ITRMAX,  IU,      IV,      I1,      I2,       HECCOM09
     9J,       JC,      JH,      JMAX,    JMAXA,   JU,      JV,      J5        HECCOM10
      COMMON   K,       KMAXA,   KMN1,    KMN2,    KP,      LLLL,    MMMM,     HECCOM11
     1MERGE,   MFTAG,   M2,      N,       NC,      NHNU,    NK,      NPC,      HECCOM12
     2NR,      NRM,     NT,      NTAPE,   NY,      N1,      N2,      N3,       HECCOM13
     3N4,      OUDC,    PABOVE,  PBLO,    P1,      P1DTS,   PRINTL,  PRINTS,   HECCOM14
     4PROB,    PHR,     PW,      RADE,    RC,      RFT,     RPTAG,   RR,       HECCOM15
     5SCUR,    SCR,     SCRE,    SCYCLE,  SGNL,    SIG,     SIGH,    SLUG,     HECCOM16
     6SN,      SPROB,   SVMAX,   SVS,     S1,      S2,      S3,      S4,       HECCOM17
     7SS,      T,       TAUDTS,  THICK,   TMAX,    TRAD,    URH,     UT,       HECCOM18
     8UU,      UVMAX,   VABOVE,  VAPE,    VBLO,    VEL,     WS,      WSA,      HECCOM19
     9XMAXX,            BLANK(49)                                              HECCOM20
      COMMON            AIX(1200),        AMX(1200),        ALAMH(1200),
     1ALAMV(1200),      B(1200),          CAP(1200),        ER(1200),
     2FIOUT(1200),      KFIT(1200),       P(1200),          SMLQ(1200),
     3THETA(1200),      U(1200),          V(1200),          XALP(1200),
     4XU(1200),         SOL1D(400),       PL(200),          Y(101),
     5DY(100),          GAMC(100),        YAMC(100),        UL(100),         HECCOM26
     6W2(100),          X(53),            DX(52),           TAU(52),         HECCOM27
     7BETAB(50),        TEMP(12),         HEAD(12)                           HECCOM28
C                                                                           DRA8038
C     THE FOLLOWING VARIABLES IN HECTIC COMMON HAVE BEEN RENAMED IN THIS
C     ROUTINE  --  CV TO CCVV , 1SEND TO 1SSSS , L TO LLLL , M TO MMMM,
C     AND XMAX TO XMAXX
C     ****************************************************************DRA8039
C                                                                          DRA8040
      DIMENSION         FLEFT(100),       PR(100),          SIGC(100)      DRA8041
      EQUIVALENCE       (FLEFT,UL),       (PL,PR)
      EQUIVALENCE      (PL(101),SIGC)
      EQUIVALENCE (S20,BLANK(12)), (ZP126,BLANK(13)), (ZP136,BLANK(14)),
     1(KMX,BLANK(15)),(WT,BLANK(16)),(ANUMBR,BLANK(19)),(LOCZ,BLANK(22))
     2,(H,BLANK(25)),(NA,BLANK(26))
      EQUIVALENCE (BACC,BLANK(47)),(SACC,BLANK(48)),(TACC,BLANK(49))
C                                                                          DRA8045
      REAL  MERGE                                                          DRA8046
C                                                                          DRA8047
C                                                                          DRA8048
      COMMON    /RAD1/     1AID,     KMAX,     1CHECK,   ISEND,            RAD1
     1L2,       L2P1,      LMAX,     M2,       MUMAX,    ISENDD,           RAD2
     21SWEEP,   LLOW,      LHI                                             RAD
      COMMON    /RAD2/     ANGL1,    ANGL2,    ASN,      CHKSUM,           RAD3
     1DELTY,    FORPI,     HELP1,    HELP2,    LMDA,     PP,               RAD4
     2PI2,      PI4,       SINP2,    TANN,     TOLER,    XMIN,             RAD5
```

```
     3XMAX,     STEF                                                    RAD6
      COMMON    /RAD3/               DUM1(1),          AA(16),          RAD7
     1          COSPHI(6),           CVIN(6),          CVOUT(6),        RAD8
     2          CH(6),               MU(6),            PHI(6),          RAD9
     3          DUM2(1),             SINLP(6),         SMU(6),          RAD10
     4          W(6),                CV(6)                              RAD11
      COMMON   /INDEX/   M,L,LP                                         RAD12
      COMMON    /DUMM/    LCHECK,   KK2,      NOBUFF                    RAD13
      COMMON    /DIS/     DV(6),    XV(6),    YV(6),    DH(6),          RAD14
     1          XH(6),    YH(6)                                         RAD15
      COMMON    /SUMCHK/             SUMB(50,6),       SUMO(50,6)       RAD16
      COMMON    /BNDT/    BTHETA(50),     RTHETA(50),     ATHETA(50)    RAD17
C
      REAL    MU , LMDA
C
      EQUIVALENCE    (FIOUT,OLDTH) , (U,ROSS) ,
     .   (CAP(1),BUFF1(1),BUFF(1),NOI) , (P(1),PLANCK(1),BUFF2(1),NOO)
      DIMENSION    OLDTH(1) , ROSS(1) , BUFF1(1) , BUFF(1) , PLANCK(1) ,
     .  BUFF2(1)
C
      EQUIVALENCE    (V(1),XA(1),EB(1))
      DIMENSION   XA(1)
      DIMENSION           EB(50,6) , EV(50,6) , ET(50,6) ,
     .            ETAIO(6,6) , ETAIT(6,6) , ETABI(6,6) , ETABT(6,6),
     .  ETABO(6,6) , ETAOI(6,6) , ETAOT(6,6) , ETAOO(6,6)
C
C
      REAL   JO
C
C   EV(IMAX,L) = THE RATE AT WHICH ENERGY IS TRANSFERRED ACROSS FACE
C     X(IMAX) IN FAN L FOR A PARTICULAR Y(J)          (ERGS/SEC)
      DO 1000 L=1,L2
      BINT = STEF/PI   * RTHETA(J) * RTHETA(J)
     .                 * RTHETA(J) * RTHETA(J)
 1000 EV(IMAX,L) = BINT * JO(J,IMAX,L,M)
      RETURN
      END
```

Subroutine TOPBND

```
      SUBROUTINE  TOPBND
C     BOUNDARY  CONDITIONS  FOR  TOP  SURFACE
C     ****************************************************************DRA8007
C        H E C T I C   C O M M O N                                     DRA8008
C     ****************************************************************DRA8009
      COMMON AHN,    AMBIEN, ANN,    ATOM,   BCATAG, BCBTAG, BCLTAG,   HECCOM01
     1BCHTAG, CAPIN,  CAPS,   CDUT,   CMXK,   CMXOM,  CNDE,   CO,       HECCOM02
     2COE,    CSTOP,  CCVV,   CYCLE,  DBGPRT, DGREY,  DHNU,   DIANTP,   HECCOM03
     3OMIN,   DNN,    DT,     DTBUGR, DTC,    DTH,    DTNA,   DT0,      HECCOM04
     4DTH,    DTR1,   DTR2,   DTUF,   DTVF,   DUMPT7, ECK,    EFRAC,    HECCOM05
     5EII,    ERRCRT, ETH,    EZERO,  FDTC,   FFA,    FFH,    FIFT,     HECCOM06
     6FUDGE,  GV,     HCB,    HCP,    HH,     HNU,    HNUP,   HNUS,     HECCOM07
     7HVB,    I,      IC,     IGOTO,  IH,     IHNU,   IMAX,   IMPTAG,   HECCOM08
     8ISSSS,  ISR,    ITAG,   ITRMAX, IU,     IV,     II,     I2,       HECCOM09
     9J,      JC,     JH,     JMAX,   JMAXA,  JU,     JV,     J5        HECCOM10
      COMMON K,      KMAXA,  KMN1,   KMN2,   KP,     LLLL,   MMMM,     HECCOM11
     1MERGE,  MFTAG,  MZ,     N,      NC,     NHNU,   NK,     NPC,      HECCOM12
     2NH,     NRM,    NT,     NTAPE,  NY,     N1,     N2,     N3,       HECCOM13
     3N4,     ODDC,   PABOVE, PBLO,   PI,     PIDTS,  PRINTL, PRINTS,   HECCOM14
     4PROB,   PRR,    PW,     RADE,   RC,     RFT,    RPTAG,  RR,       HECCOM15
     5SCUR,   SCR,    SCHE,   SCYCLE, SGNL,   SIG,    SIGH,   SLUG,     HECCOM16
     6SH,     SPROB,  SVMAX,  SVS,    S1,     S2,     S3,     S4,       HECCOM17
     7SS,     T,      TAUDTS, THICK,  TMAX,   TRAD,   URR,    UT,       HECCOM18
     8UU,     UVMAX,  VABOVE, VAPE,   VBLO,   VEL,    WS,     WSA,      HECCOM19
     9XMAXX,          BLANK(49)                                         HECCOM20
      COMMON          AIX(1200),      ANK(1200),      ALAMH(1200),
     1ALAMV(1200),    B(1200),        CAP(1200),      ER(1200),
     2FIOUT(1200),    KFIT(1200),     P(1200),        SMLO(1200),
     3THETA(1200),    U(1200),        V(1200),        XALP(1200),
     4XU(1200),       SOLID(400),     PL(200),        Y(101),
     5UT(100),        GAMC(100),      YAMC(100),      UL(100),          HECCOM26
     6Z(100),         X(53),          DX(52),         TAU(52),          HECCOM27
     7BETAB(50),      TEMP(12),       HEAD(12)                          HECCOM28
C                                                                       DRA8038
C     THE FOLLOWING VARIABLES IN HECTIC COMMON HAVE BEEN RENAMED IN THIS
C     ROUTINE  --  CV TO CCVV , ISEND TO ISSSS , L TO LLLL , M TO MMMM,
C     AND XMAX TO XMAXX
C     ****************************************************************DRA8039
C                                                                       DRA8040
      DIMENSION       FLEFT(100),     PR(100),        SIGC(100)         DRA8041
      EQUIVALENCE     (FLEFT,UL),     (PL,PR)
      EQUIVALENCE     (PL(101),SIGC)
      EQUIVALENCE (S20,BLANK(12)), (ZP126,BLANK(13)), (ZP136,BLANK(14)),
     1(KMX,BLANK(15)),(WT,BLANK(16)),(ANUMBR,BLANK(19)),(LOCZ,BLANK(22))
     2,(H,BLANK(25)),(NA,BLANK(26))
      EQUIVALENCE (BACC,BLANK(47)),(SACC,BLANK(48)),(TACC,BLANK(49))
C                                                                       DRA8045
      REAL  MERGE                                                       DRA8046
C                                                                       DRA8047
C                                                                       DRA8048
      COMMON     /RAD1/     IAIU,      KMAX,      ICHECK,    ISEND,     RAD1
     1L2,        L2P1,      LMAX,      M2,        MUMAX,     ISENDD,    RAD2
     2ISWEEP,    LLOW,      LHI                                         RAD
      COMMON     /RAD2/     ANGL1,     ANGL2,     ASN,       CHKSUM,    RAD3
     1DELTY,     FORPI,     HELP1,     HELP2,     LMDA,      PP,        RAD4
```

```
     2PI2,      PI4,       SINP2,     TANN,      TOLER,     XMIN,          RAD5
     3XMAX,     STEF                                                       RAD6
      COMMON    /RAD3/                DUM1(1),              AA(16),        RAD7
     1          COSPHI(6),            CVIN(6),              CVOUT(6),      RAD8
     2          CH(6),                MU(6),                PHI(6),        RAD9
     3          DUM2(1),              SINLP(6),             SMU(6),        RAD10
     4          W(6),                 CV(6)                                RAD11
      COMMON   /INDEX/   M,L,LP                                            RAD12
      COMMON    /DRUMM/   LCHECK,    KK2,       NOBUFF                     RAD13
      COMMON    /DIS/     DV(6),     XV(6),     YV(6),      DH(6),         RAD14
     1          XH(6),    YH(6)                                            RAD15
      COMMON    /SUMCHR/              SUMB(50,6),           SUMO(50,6)     RAD16
      COMMON    /BNDT/    BTHETA(50),        RTHETA(50),     ATHETA(50)    RAD17
C
      REAL    MU , LMDA
C
      EQUIVALENCE    (FIOUT,OLDTH) , (U,ROSS) ,
     .    (CAP(1),BUFF1(1),BUFF(1),NOI) , (P(1),PLANCK(1),BUFF2(1),NOO)
      DIMENSION   OLDTH(1) , ROSS(1) , BUFF1(1) , BUFF(1) , PLANCK(1) ,
     .  BUFF2(1)
C
      EQUIVALENCE    (V(1),XA(1),EB(1))
      DIMENSION   XA(1)
      DIMENSION             EB(50,6) , EV(50,6) , ET(50,6) ,
     .          ETAIO(6,6) , ETAIT(6,6) , ETABI(6,6) , ETABT(6,6),
     .  ETABO(6,6) , ETAOI(6,6) , ETAOT(6,6) , ETAOO(6,6)
C
C
      REAL   JB
C
      IF (BCATAG .LT. 0.)  GO TO 1100
      DO 1000 I=1,IMAX
      DO 900 L=0,LMAX
 900  AA(L) = SINLP(L)*X(I)
      DO 1000 L=1,LMAX
      BINT = STEF/PI  * ATHETA(I) * ATHETA(I)
     .                * ATHETA(I) * ATHETA(I)
1000 EB(I,L) = BINT * JB(JMAX,I,L,M)
1100 RETURN
      END
```

This page intentionally left blank.

Subroutine TDRAD (ADI)

```
      SUBROUTINE TDRAD                                                  TDR2000
C                                                                       TDR2001
C     ******************************************************************TDR2002
C          H E C T I C     C O M M O N                                  TDR2003
C     ******************************************************************TDR2004
C                                                                       TDR2005
      COMMON AHN,    AMBIEN, ANN,    ATOM,   BCATAG, BCBTAG, BCLTAG,    HECCOM01
     1BCRTAG, CAPIN, CAPS,   CDUT,   CMXK,   CMXOM,  CNOE,   CO,        HECCOM02
     2COE,    CSTOP, CV,     CYCLE,  DBGPRT, DGREY,  DHNU,   DIANTP,    HECCOM03
     3DMIN,   DNN,   DT,     DTBUGR, DTC,    DTH,    DTNA,   DTQ,       HECCOM04
     4DTR,    DTR1,  DTR2,   DTUF,   DTVF,   DUMPT7, ECK,    EFRAC,     HECCOM05
     5E11,    ERRCRT, ETH,   EZERO,  FDTC,   FFA,    FFB,    FIFT,      HECCOM06
     6FUDGE,  GV,    HCB,    HCP,    HH,     HNU,    HNUP,   HNUS,      HECCOM07
     7HVB,    I,     IC,     IGOTO,  IH,     IHNU,   IMAX,   IMPTAG,    HECCOM08
     8ISEND,  ISR,   ITAG,   ITRMAX, IU,     IV,     I1,     I2,        HECCOM09
     9J,      JC,    JH,     JMAX,   JMAXA,  JU,     JV,     J5         HECCOM10
      COMMON K,      KMAXA,  KMN1,   KMN2,   KP,     L,      M,         HECCOM11
     1MERGE,  MFTAG, MZ,     N,      NC,     NHNU,   NK,     NPC,       HECCOM12
     2NR,     NRM,   NT,     NTAPE,  NY,     N1,     N2,     N3,        HECCOM13
     3N4,     OUDC,  PABOVE, PBLO,   PI,     PIDTS,  PRINTL, PRINTS,    HECCOM14
     4PROB,   PRR,   PW,     RADE,   RC,     RFT,    RPTAG,  RR,        HECCOM15
     5SCDR,   SCR,   SCRE,   SCYCLE, SGNL,   SIG,    SIGH,   SLUG,      HECCOM16
     6SN,     SPROB, SVMAX,  SVS,    S1,     S2,     S3,     S4,        HECCOM17
     7S5,     T,     TAUDTS, THICK,  TMAX,   TRAD,   URR,    UT,        HECCOM18
     8UU,     UVMAX, VABOVE, VAPE,   VBLO,   VEL,    WS,     WSA,       HECCOM19
     9XMAX,          BLANK(49)                                          HECCOM20
      COMMON         A1X(1200),      AMX(1200),      ALAMH(1200),       HECCOM21
     1ALAMV(1200),   B(1200),        CAP(1200),      ER(1200),          HECCOM22
     2FIOUT(1200),   KFIT(1200),     P(1200),        SMLQ(1200),        HECCOM23
     3THETA(1200),   U(1200),        V(1200),        XALP(1200),        HECCOM24
     4XB(1200),      SOLID(400),     PL(200),        Y(101),            HECCOM25
     5DY(100),       GAMC(100),      YAMC(100),      UL(100),           HECCOM26
     6W2(100),       X(53),          DX(52),         TAU(52),           HECCOM27
     7BETAB(50),     TEMP(12),       HEAD(12)                           HECCOM28
C                                                                       TDR2033
C     ******************************************************************TDR2034
C                                                                       TDR2035
      DIMENSION      FLEFT(100),     PR(100),        SIGC(100)          TDR2036
      DIMENSION      Z(200),         IZ(200)                            TDR2037
      EQUIVALENCE    (FLEFT,UL),     (PL,PR),        (Z,IZ,AHN)         TDR2038
      EQUIVALENCE  (PL(101), SIGC)                                      TDR2039
      EQUIVALENCE  (BLANK(1),ILUMI),(BLANK(2),ISL)
      EQUIVALENCE  (ZP,BLANK(12)), (ZP12N,BLANK(13)), (ZP13N,BLANK(14)),
     1(KMX,BLANK(15)),(W,BLANK(16)),(ANUMBR,BLANK(19)),(LOCZ,BLANK(22))
     2,(Q,BLANK(25)),(HA,BLANK(26))
C                                                                       TDR2040
      REAL  MERGE                                                       TDR2041
C                                                                       TDR2042
C                                                                       TDR2043
      DIMENSION  POR(50), POZ(100), ROR(50), ROZ(100)                   TDR2044
      DIMENSION OLDTH(1),PLANCK(1),ROSS(1),XA(1)
      DIMENSION KXA(1)                                                  TDR2046
      DIMENSION  KXB(1)                                                 TDR2047
      EQUIVALENCE (OLDTH,FIOUT),(PLANCK,P),(ROSS,U),(KXA,XA,V)
     1 , (KXB,XB)
C                                                                       TDR2050
```

```
C                                                                          TDR2051
      DIMENSION XC(1200),XGAM(1200),XD(1200),XG(1200),ERAD(1200),         TDR2052
     1          BETAA(50),BETAR(50)                                        TDR2053
      DIMENSION EH(1)                                                      TDR2054
      EQUIVALENCE (XG,EH)
      DIMENSION XGI(100),XUI(100)
      DATA XLAM /7.5/                                                      TDR2055
C
C     INITIALIZE
C
      IDBGP = DBGPRT
      FOO=0.0
      KMAX=KMAXA-1                                                         TDR2056
      ITER = 1
      IF (MFTAG.EQ.0) DHNU=1.                                              TDR2057
      NVEZ=1                                                               TDR2058
      NY=NVEZ                                                              TDR2059
      IF (ITAG.EQ.0) NVEZ=2                                                TDR2060
      VEZ=NVEZ                                                             TDR2061
C
C     CALCULATE GEOMETRY FACTORS
C
      DO 10 I=1,IMAX                                                       TDR2065
      PUR(I)=PI/TAU(I)                                                     TDR2066
   10 RUR(I+1)=4.0E10*X(I)/(X(I+1)-X(I-1))                                 TDR2067
      DO 20 J=1,JMAX                                                       TDR2068
      PUZ(J)=1./DY(J)                                                      TDR2069
   20 RUZ(J+1)=2.0E10/(Y(J+1)-Y(J-1))                                      TDR2070
C
C     FIND HIGHEST TEMPERATURE
C
      THTAMX=.025                                                          TDR2063
      DO 30 K=2,KMAX                                                       TDR2071
      IF (JMN(KFIT(K),2).NE.1) GO TO 30                                    TDR2072
      IF (THETA(K).LE.THTAMX) GO TO 30                                     TDR2073
      THTAMX=THETA(K)                                                      TDR2074
   30 CONTINUE                                                             TDR2075
      THTAMX=AMAX1(THTAMX,BCLTAG,BCRTAG,BCBTAG,BCATAG)                     TDR2076
C
C     REENTRY POINT FOR SECOND TEMPERATURE ITERATION                       TDR2077
C
   40 DO 50 K=2,KMAX                                                       TDR2078
      XA(K)=0.                                                             TDR2079
      XB(K)=0.                                                             TDR2080
      XC(K)=0.                                                             TDR2081
      XALP(K)=0.                                                           TDR2082
      ER(K)=0.0                                                            TDR2083
   50 XGAM(K)=0.                                                           TDR2084
C                                                                          TDR2085
C                                                                          TDR2086
C     *****************************************************************TDR2087
C                                                                     *TDR2088
C               B E G I N   F R E Q U E N C Y   L O O P               *TDR2089
C                                                                     *TDR2090
C     *****************************************************************TDR2091
```

```
C                                                                       TDR2092
C               SET UP MAX FREQ BOUNDARY                                TDR2093
C                                                                       TDR2094
      HNUP=1.0E6                                                        TDR2095
      HNUP4=1.0E24                                                      TDR2096
      IF (MFTAG.EQ.0) GO TO 160                                         TDR2097
C
C     MULTIFREQUENCY CALCULATION
C
   60 IHNU=IHNU+1                                                       TDR2098
      CALL KAPPA                                                        TDR2099
      HNU4=HNU**4                                                       TDR2100
      DHNU=HNUP-HNU                                                     TDR2101
C                                                                       TDR2102
C     MERGE GROUPS WITH HNU MORE THAN (MERGE) TIMES LARGEST THETA       TDR2103
C                                                                       TDR2104
      IF (MERGE.GT.0.0) GO TO 70                                        TDR2105
      S1 = 8.0225                                                       TDR2106
      GO TO 845
   70 IF (THTAMX-HNU/MERGE) 100,80,80                                   TDR2108
   80 IF (IHNU-1) 90,210,120                                            TDR2109
   90 S1 = 8.0235                                                       TDR2110
      GO TO 845
C                                                                       TDR2112
C               REJECT TAPE IF MORE THAN HALF OF GROUPS MERGE           TDR2113
C                                                                       TDR2114
  100 IF (IHNU+IHNU-NHNU) 120,110,110                                   TDR2115
  110 IF (AMOD(MERGE,1.).EQ.0.5) GO TO 120                              TDR2116
      S1 = 8.0250                                                       TDR2117
      GO TO 845
C                                                                       TDR2131
C               FORM NUMERATORS AND DENOMINATORS OF MERGED KAPPAS       TDR2132
C          STORE IN XA, XB, XC, XALP
C                                                                       TDR2133
  120 DO 130 K=2,KMAX                                                   TDR2119
      IF (JMR(KFIT(K),2).NE.1) GO TO 130                                TDR2120
      T4=THETA(K)**4                                                    TDR2121
      BETA=HNU/THETA(K)                                                 TDR2122
      BETAP=HNUP/THETA(K)                                               TDR2123
      DFB=PLNKUT(BETA,BETAP)                                            TDR2124
      IF (DFB.EQ.0.) GO TO 130                                          TDR2125
      TEMP(1)=DFB*T4                                                    TDR2126
      EMB1=EXP(-BETA)                                                   TDR2127
      EMB2=EXP(-BETAP)                                                  TDR2128
      TEMP(2)=DFB+0.0384974/T4*(HNU4/(1.0-EMB1)*EMB1-HNUP4/(1.0-EMB2)*EMTDR2129
     1B2)                                                               TDR2130
      XA(K)=XA(K)+TEMP(1)                                               TDR2134
      XB(K)=XB(K)+TEMP(2)                                               TDR2135
      XC(K)=XC(K)+PLANCK(K)*TEMP(1)                                     TDR2136
      XALP(K)=XALP(K)+TEMP(2)/ROSS(K)                                   TDR2137
  130 CONTINUE                                                          TDR2138
      HNUP=HNU                                                          TDR2139
      HNUP4=HNU4                                                        TDR2140
      IF (THTAMX-HNU/MERGE) 60,140,140                                  TDR2141
C                                                                       TDR2142
```

```
              FORM MERGED KAPPAS
140 DO 170 K=2,KMAX
    IF (XA(K)) 150,170,160
150 S1 = 8.0320
    GO TO 845
160 ROSS(K)=XB(K)/XALP(K)
    PLANCK(K)=XC(K)/XA(K)
170 CONTINUE
    HNU=1.0E6
    HNUP4=1.0E24
    DHNU=HNUP-HNU
    GO TO 230
              MONOFREQUENCY CALCULATION
180 NHNU=1
    IHNU=1
    HNU=.001
    DO 190 K=2,KMAX
    IF (JMR(KFIT(K),2).NE.1) GO TO 190
    B(K)=THETA(K)**4
190 CONTINUE
    GO TO 230
              TYPICAL GROUP CALCULATION OF SOURCES
200 IHNU=IHNU+1
    CALL KAPPA
    DHNU=HNUP-HNU
    HNU4=HNU**4
210 DO 220 K=2,KMAX
    IF (JMR(KFIT(K),2).NE.1) GO TO 220
    DFB=PLNKUT(HNU/THETA(K),HNUP/THETA(K))
    B(K)=DFB*THETA(K)**4
220 CONTINUE
              SET BLACKBODY CONDITIONS
    INNER CYLINDRICAL RADIUS NOT ASSUMED ZERO
230 BBL=0.
    IF (BCLTAG.LE.0.0) GO TO 240
    DFB=PLNKUT(HNU/BCLTAG,HNUP/BCLTAG)
    BBL=BCLTAG**4*DFB
240 BBR=0.
    IF (BCRTAG.LE.0.0) GO TO 250
    DFB=PLNKUT(HNU/BCRTAG,HNUP/BCRTAG)
    BBR=BCRTAG**4*DFB
250 BBB=0.0
    IF (BCBTAG.LE.0.0) GO TO 260
    DFB=PLNKUT(HNU/BCBTAG,HNUP/BCBTAG)
    BBB=BCBTAG**4*DFB
260 BBA=0.0
    IF (BCATAG .LE. 0.) GO TO 290
```

```
TDR2143
TDR2144
TDR2145
TDR2146
TDR2147
TDR2149
TDR2150
TDR2151
TDR2152
TDR2153
TDR2154
TDR2155
TDR2156
TDR2157
TDR2158
TDR2159
TDR2160
TDR2165
TDR2161
TDR2162
TDR2163
TDR2164
TDR2166
TDR2167
TDR2168
TDR2169
TDR2170
TDR2171
TDR2172
TDR2173
TDR2174
TDR2175
TDR2176
TDR2177
TDR2178
TDR2179
TDR2180
TDR2181
TDR2182
TDR2183
TDR2184
TDR2185
TDR2186
TDR2187
TDR2188
TDR2189
TDR2190
TDR2191
TDR2192
TDR2193
TDR2194
TDR2195
```

```
      DFB=PLNKUT(HNU/BCATAG,HNUP/BCATAG)                                TDR2197
      BBA=BCATAG**4*DFB                                                 TDR2198
                                                                        TDR2203
               FORM ROSSELAND AND PLANCK OPTICAL DEPTHS                 TDR2204
                                                                        TDR2205
  290 DO 340 I=1,IMAX                                                   TDR2207
      K=I+1                                                             TDR2208
      M=K-IMAX                                                          TDR2209
      W2(I)=0.0                                                         TDR2210
      DO 340 J=1,JMAX                                                   TDR2211
      FACTOR=1.                                                         TDR2212
      IF (RPTAG.NE.0.0) FACTOR=ROSS(K)/PLANCK(K)                        TDR2213
      ROSS(K)=AMAX1(ROSS(K)+AMX(K)/(TAU(I)*DY(J)),1.E-20)
      PLANCK(K)=AMAX1(PLANCK(K)+FACTOR*AMX(K)/(TAU(I)*DY(J)),1.E-20)

      FORM LAMBDA, MEAN FREE PATH AT CELL EDGES                         TDR2216

      IF (I.EQ.1) GO TO 310                                             TDR2217
      DTAUR=ROSS(K)*DX(I)                                               TDR2220
      DTAUL=ROSS(K-1)*DX(I-1)                                           TDR2221
      IF (ABS(DTAUR-DTAUL)/(DTAUR+DTAUL).LE.ODDC) GO TO 300             TDR2222
      ALAMV(K)=AMAX1(1./ROSS(K),1./ROSS(K-1))                           TDR2223
      GO TO 310                                                         TDR2224
  300 ALAMV(K)=(DX(I)+DX(I-1))/(DTAUR+DTAUL)                            TDR2225
  310 IF (J.EQ.1) GO TO 330                                             TDR2226
      DTAUA=ROSS(K)*DY(J)                                               TDR2227
      DTAUB=ROSS(M)*DY(J-1)                                             TDR2228
      IF (ABS(DTAUA-DTAUB)/(DTAUA+DTAUB).LE.ODDC) GO TO 320             TDR2229
      ALAMH(K)=AMAX1(1./ROSS(K),1./ROSS(M))                             TDR2230
      GO TO 330                                                         TDR2231
  320 ALAMH(K)=(DY(J)+DY(J-1))/(DTAUA+DTAUB)                            TDR2232
  330 M=M+IMAX                                                          TDR2233
      K=K+IMAX                                                          TDR2234
  340 CONTINUE                                                          TDR2235

      BEGIN NONEQUILIBRIUM DIFFUSION TREATMENT                          TDR2237
                                                                        TDR2238
      FORM MATRIX ELEMENTS XA, XB, XC, XD, XALP, XGAM                   TDR2239
      BOUNDARY CONDITION MODIFIES XB AND XD.                            TDR2241
      THIS CODING ASSUMES DX(I) = X(I+1) - X(I), SAME FOR Y.            TDR2242

      DO 440 I=1,IMAX                                                   TDR2243
      K=I+1                                                             TDR2244
      N=K+IMAX                                                          TDR2245
      DO 440 J=1,JMAX                                                   TDR2246
      XB(K)=0.                                                          TDR2247
      XD(K)=0.                                                          TDR2248
      IF (I.EQ.1) GO TO 350                                             TDR2249
      XA(K)=-PUR(I)*RUR(I)*ALAMV(K)                                     TDR2250
      GO TO 360                                                         TDR2251
  350 XA(K)=0.0                                                         TDR2252
      INNER CYLINDRICAL RADIUS ASSUMED ZERO                             TDR2253
  360 IF (I.EQ.IMAX) GO TO 370                                          TDR2254
      XC(K)=-PUR(I)*RUR(I+1)*ALAMV(K+1)                                 TDR2255
      GO TO 380                                                         TDR2256
```

```
370 XC(K)=0.0                                                  TDR2257
    IF (BCRTAG.LT.0.0) GO TO 380                               TDR2258
    BETAR(J)=1.0/(2.-EXP(-DX(I)*SQRT(0.75*ROSS(K)*PLANCK(K)))) TDR2259
    XB(K)=XB(K)+PUR(I)*3.E10*X(I)*BETAR(J)                     TDR2260
    XD(K)=XD(K)-4.104E12*PUR(I)*X(I+1)*BBR                     TDR2261
380 IF (J.EQ.1) GO TO 390                                      TDR2262
    XALP(K)=-PUZ(J)*RUZ(J)*ALAMH(K)                            TDR2263
    GO TO 400                                                  TDR2264
390 XALP(K)=0.0
    IF (BCBTAG.LT.0.0) GO TO 400
    BETAB(I)=1.0/(2.-EXP(-DY(1)*SQRT(0.75*ROSS(K)*PLANCK(K)))) TDR2267
    XB(K)=XB(K)+PUZ(1)*1.5E10*BETAB(I)                         TDR2268
    XD(K)=XD(K)-PUZ(1)*2.052E12*BBB                            TDR2269
400 IF (J.EQ.JMAX) GO TO 410                                   TDR2270
    XGAM(K)=-PUZ(J)*RUZ(J+1)*ALAMH(N)                          TDR2271
    GO TO 420                                                  TDR2272
410 XGAM(K)=0.0                                                TDR2273
    IF (BCATAG.LT.0.0) GO TO 420                               TDR2274
    BETAA(I)=1.0/(2.-EXP(-DY(J)*SQRT(0.75*ROSS(K)*PLANCK(K)))) TDR2275
    XB(K)=XB(K)+PUZ(J)*1.5E10*BETAA(I)                         TDR2276
    XD(K)=XD(K)-PUZ(J)*2.052E12*BBA                            TDR2277
420 XB(K) = XB(K) + 3.E10*PLANCK(K)                            TDR2278
    XD(K)=XD(K)-4.104E12*PLANCK(K)*B(K)                        TDR2279

    INITIALIZE ERAD

    ERAD(K)=B(K)*137.                                          TDR2282
430 K=K+IMAX                                                   TDR2291
    N=N+IMAX                                                   TDR2292
440 CONTINUE                                                   TDR2293

    OPTIONAL PRINTOUT

    IF(AMOD(CYCLE,PRINTL).NE.0) GO TO 442
    WRITE (6,870) CYCLE,T,DT
    CALL SPEW(XA,IMAX,JMAX,2HXA)
    CALL SPEW(XB,IMAX,JMAX,2HXB)
    CALL SPEW(XC,IMAX,JMAX,2HXC)
    CALL SPEW(XD,IMAX,JMAX,2HXD)
    CALL SPEW(XGAM,IMAX,JMAX,4HXGAM)
    CALL SPEW(XALP,IMAX,JMAX,4HXALP)
    CALL SPEW(ROSS,IMAX,JMAX,4HROSS)
    CALL SPEW(PLANCK,IMAX,JMAX,6HPLANCK)
    CALL SPEW(B,IMAX,JMAX,1HB)
                                                               TDR2294
    START ADI SOLUTION

442 ERADM = 0.
    ERRORM = 0.
    DO 460 J=1,JMAX

    CALCULATE XDI AND XGI FOR VERTICAL SWEEP

    M = (J-1)*IMAX + 2
    BEE = (1. + XLAM)*XB(M)  -XA(M) - XC(M)
```

```
      IF(BEE.NE.0.) GO TO 443
      XGI(1) = 0.
      XDI(1) = 0.
      GO TO 444
  443 DEE = -XD(M) - XALP(M)*(ERAD(M-IMAX) - ERAD(M))
     1           - XGAM(M)*(ERAD(M+IMAX) - ERAD(M))
     2           + XLAM*XB(M)*ERAD(M)
      XGI(1) = -XC(M)/BEE
      XDI(1) = DEE/BEE
  444 DO 446 I=2,IMAX
      K = (J-1)*IMAX + I + 1
      BEE = (1. + XLAM)*XB(K) - XA(K) - XC(K)
      DEE = -XD(K) - XALP(K)*(ERAD(K-IMAX) - ERAD(K))
     1           - XGAM(K)*(ERAD(K+IMAX) - ERAD(K))
     2            + XLAM*XB(K)*ERAD(K)
      DENOM = XA(K)*XGI(I-1) + BEE
      IF(DENOM.NE.0.) GO TO 445
      XGI(I) = 0.
      XDI(I) = 0.
      GO TO 446
  445 XGI(I) = -XC(K)/DENOM
      XDI(I) = (DEE - XA(K)*XDI(I-1))/DENOM
  446 CONTINUE

      CALCULATE EH (HALF STEP ERAD)

      EH(K) = XDI(IMAX)
      IMAXM1 = IMAX - 1
      DO 447 I=1,IMAXM1
      K = K - 1
      II = IMAX - I
      EH(K) = XGI(II)*EH(K+1) + XDI(II)
  447 CONTINUE
  460 CONTINUE
      DO 500 I=1,IMAX

      CALCULATE XDI AND XGI FOR HORIZONTAL SWEEP

      M = I + I
      BEE = (1. + XLAM)*XB(M) - XALP(M) - XGAM(M)
      IF(BEE.NE.0.) GO TO 470
      XGI(1) = 0.
      XDI(1) = 0.
      GO TO 480
  470 DEE = -XD(M) - XA(M)*(EH(M-1) - EH(M))
     1           - XC(M)*(EH(M+1) - EH(M))
     2           + XLAM*XB(M)*EH(M)
      XGI(1) =-XGAM(M)/BEE
      XDI(1) = DEE/BEE
  480 DO 485 J=2,JMAX
      K = (J-1)*IMAX + I + 1
      BEE = (1. + XLAM)*XB(K) - XALP(K) - XGAM(K)
      DEE = - XD(K) - XA(K)*(EH(K-1) - EH(K))
     1           - XC(K)*(EH(K+1) - EH(K))
     2            + XLAM*XB(K)*EH(K)
```

```
      DENOM = XALP(K)*XGI(J-1) + BEE
      IF(DENOM.NE.0.) GO TO 482
      XGI(J) = 0.
      XDI(J) = 0.
      GO TO 485
  482 XGI(J) = - XGAM(K)/DENOM
      XDI(J) = (DEE - XALP(K)*XDI(J-1))/DENOM
  485 CONTINUE

      CALCULATE ERAD

      ERADP = ERAD(K)
      ERAD(K) = XDI(JMAX)
      ERADM = AMAX1(ERADM,ERAD(K))
      ERRORM = AMAX1(ERRORM,ABS(ERAD(K) - ERADP))
  486 JMAXM1 = JMAX - 1
      DO 490 J=1,JMAXM1
      K = K - IMAX
      JJ = JMAX - J
      ERADP = ERAD(K)
      ERAD(K) = XGI(JJ)*ERAD(K+IMAX) + XDI(JJ)
      ERADM = AMAX1(ERADM,ERAD(K))
      ERRORM = AMAX1(ERRORM,ABS(ERAD(K) - ERADP))
  490 CONTINUE
  500 CONTINUE

      SET NEGATIVE ERAD TO 0.

      DO 545 K=2,KMAX
      IF(ERAD(K).LT.0.) ERAD(K) = 0.                          TDR2336
  545 CONTINUE
      ERROR = ERRORM/ERADM

      OPTIONAL PRINTOUT

      IF(AMOD(CYCLE,PRINTS) .NE. 0.) GO TO 560
      WRITE (6,1010) ITER,ERROR
      IF(MOD(ITER,IDBGP) .NE. 1) GO TO 560
      CALL SPEW(ERAD,IMAX,JMAX,4HERAD)

      TEST FOR CONVERGENCE

  560 ITER = ITER + 1
      IF(ITER .LE. ITRMAX) GO TO 570
      CALL SPEW(ERAD,IMAX,JMAX,4HERAD)
      S1 = 8.0560
      GO TO 845
  570 IF (ERROR.GT.ERRCRT) GO TO 442

      CALCULATE FLUXES

      DO 640 I=1,IMAX                                         TDR2354
      K=I+1                                                   TDR2355
      M=K-IMAX                                                TDR2356
      DO 640 J=1,JMAX                                         TDR2357
```

```
      IF (J.EQ.1) GO TO 580                                               TDR2358
      XA(K)=HUZ(J)*ALAMH(K)*(ERAD(M)-ERAD(K))                             TDR2359
      GO TO 610                                                           TDR2360
  580 IF (BCHTAG) 590,600,600                                             TDR2361
  590 XA(K)=0.0                                                           TDR2362
      GO TO 610                                                           TDR2363
  600 XA(K)=-2.052E12*BHH-1.5E10*ERAD(K)*BETAH(I)                         TDR2364
      FOO=FOO+XA(K)*TAU(I)                                                TDR2365
  610 IF (I.EQ.1) GO TO 620                                               TDR2366
      XB(K)=HUH(I)*ALAMV(K)*(ERAD(K-1)-ERAD(K))                           TDR2367
      GO TO 630                                                           TDR2368
  620 XB(K)=0.0                                                           TDR2369
  630 K=K+IMAX                                                            TDR2370
      M=M+IMAX                                                            TDR2371
  640 CONTINUE                                                            TDR2372
      DO 720 I=1,IMAX                                                     TDR2373
      K=I+1                                                               TDR2374
      M=K+IMAX                                                            TDR2375
      DO 720 J=1,JMAX                                                     TDR2376
      FABV=XA(M)                                                          TDR2377
      IF (J.NE.JMAX) GO TO 680                                            TDR2378
      IF (BCATAG) 650,660,660                                             TDR2379
  650 FABV=0.0                                                            TDR2380
      GO TO 670                                                           TDR2381
  660 FABV=-2.052E12*BUA+1.5E10*ERAD(K)*BETAA(I)                          TDR2382
      FOO=FOO-FABV*TAU(I)                                                 TDR2383
  670 W2(I)=FABV+W2(I)                                                    TDR2384
  680 FRT=XB(K+1)                                                         TDR2385
      IF (I.NE.IMAX) GO TO 710                                            TDR2386
      IF (BCRTAG) 690,700,700                                             TDR2387
  690 FRT=0.0                                                             TDR2388
      GO TO 710                                                           TDR2389
  700 FRT=(-2.052E+12*BBR+3.0E+10*ERAD(K)*BETAR(J))*X(IMAX)               TDR2390
      FOO=FOO-FRT*PI*DY(J)                                                TDR2391
  710 ER(K)=ER(K)+(XA(K)-FABV)*TAU(I)+(XB(K)-FRT)*PI*DY(J)                TDR2392
      K=K+IMAX                                                            TDR2393
      M=M+IMAX                                                            TDR2394
  720 CONTINUE                                                            TDR2395
                                                                          TDR2396
      OPTIONAL PRINTOUT
                                                                          TDR2398
      IF (AMOD(CYCLE,PRINTL) .NE. 0.) GO TO 750
      CALL SPEW(XB,IMAX,JMAX,6HH FLUX)
      CALL SPEW(XA,IMAX,JMAX,6HV FLUX)
      CALL SPEW(ER,IMAX,JMAX,2HER)
                                                                          TDR2404
      ADVANCE FREQ, STORE EMERGENT FLUX, TEST FOR COMPLETION OF GROUPS    TDR2405
                                                                          TDR2406
  750 HNUP=HNU                                                            TDR2409
      HNUP4=HNU4                                                          TDR2410
      IF (IHNU-NHNU) 200,770,760                                          TDR2411
                                                                         *TDR2413
                  E N D   F R E Q U E N C Y   L O O P                    *TDR2414
                                                                         *TDR2415
  760 S1 = 0.1050                                                         TDR2417
```

```
      GO TO 845

      ITERATE ON TEMPERATURE                                              TDR2419

  770 GO TO (780,800), NVEZ                                               TDR2420
  780 NVEZ=2                                                              TDR2421
      VEZ=NVEZ                                                            TDR2422
      NY=NVEZ                                                             TDR2423
      IHNU=0                                                              TDR2424
      DO 790 K=2,KMAX                                                     TDR2425
      WORK, SOURCE TERMS OMITTED                                          TDR2426
      IF (JMR(KFIT(K),2).NE.1) GO TO 790                                  TDR2427
      OLDTH(K)=THETA(K)                                                   TDR2428
      E=AIX(K)+ER(K)*DT/AMX(K)                                            TDR2429
      SV=TAU(1)*DY(J)/AMX(K)
      CALL ES (SV,E,TEMP(1),TEMP(2),66)                                   TDR2431
      THETA(K)=0.5*(THETA(K)+TEMP(1))                                     TDR2432
  790 CONTINUE                                                            TDR2433
      IF (MFTAG.EQ.0) CALL KAPPA                                          TDR2434
      GO TO 40                                                            TDR2435
  800 IF (ITAG.EQ.0) GO TO 820                                            TDR2436
      DO 810 K=2,KMAX                                                     TDR2437
  810 THETA(K)=OLDTH(K)                                                   TDR2438

      CHANGE INTERNAL ENERGIES

  820 TEMP(1)=1.                                                          TDR2440
      DO 830 K=2,KMAX                                                     TDR2441
      IF(JMR(KFIT(K),2).NE.1) GO TO 830
      DE=ER(K)*DT/AMX(K)                                                  TDR2443
      Q=SLUG*AIX(K)                                                       TDR2444
      IF(ABS(DE) .LT. ABS(Q)) GO TO 830
      TEMP(1)=AMIN1(Q/ABS(DE),TEMP(1))                                    TDR2446
  830 CONTINUE                                                            TDR2450
      DTEMP = DT*TEMP(1)
      IF(DTEMP .GT. FFB) GO TO 850
  840 WRITE (6,960) NC,DT                                                 TDR2453
      CALL SPEW(ER,IMAX,JMAX,2HER)
      S1 = 8.1068                                                         TDR2455
  845 ISEND = 2
      READ (4) FIOUT,P,U,V
      CALL EDIT
  850 T = T - DT + DTEMP
      DT = DTEMP
      ETH = ETH + FOO*DT
      WRITE (6,970) NC,DT                                                 TDR2464
      READ (4) FIOUT,P,U,V
      REWIND 4
      RETURN                                                              TDR2466
                                                                          TDR2467
  870 FORMAT (7H1CYCLE F6.0,7H  TIME 1PE13.6,5H  DT 1PE13.6/)             TDR2468
  950 FORMAT (4H K =I4,5H DE =1PE11.4,6H AMX =E11.4,4H Q =E11.4,6H AIX =TDR2478
     1E11.4)                                                              TDR2479
  960 FORMAT (52H1INTERNAL ENERGY GAIN PER UNIT TIME DUE TO RADIATION,8HTDR2480
     1 CYCLE =I5,5H DT =1PE11.4)                                          TDR2481
```

```
 970 FORMAT (19H0   FLUX FOR CYCLE I4,7H   DT =1PE10.3)          TDR2482
1010 FORMAT(1H0,9HITERATION,I6,10X,21HMAX. ERROR/ MAX. ERAD,F16.8)      TDR2483
     END
```

Modifications to Subroutine TDRAD (ADI) for OLIPHANT

```
      FOR OLIPHANT

     REMOVE TDR2054 - TDR2055

     REPLACE TDR2278 - TDR2279 BY

420  XB(K)=XB(K)+3.E10*PLANCK(K)-XA(K)-XC(K)-XALP(K)-XGAM(K)            TDR2278
     XD(K)=XD(K)-4.104E12*PLANCK(K)*B(K)                                TDR2279

     REPLACE TDR2294 - TDR2300 BY

     FIRST PASS -- P. 16, EQS. 47-49 OF OLIPHANT                        TDR2294

490  DO 445 K=2,KMAX
     M=K-IMAX                                                           TDR2296
     XB(K)=XB(K)-XALP(K)*XGAM(M)-XA(K)*XC(K-1)                          TDR2297
     XGAM(K)=XGAM(K)*CMXK/XB(K)                                         TDR2298
     XC(K)=XC(K)*CMXK/XB(K)                                             TDR2299
445  CONTINUE

     SECOND PASS -- EQS. 50, 51 OF OLIPHANT                             TDR2312

495  DO 510 K=2,KMAX
     M=K-IMAX                                                           TDR2314
     N=K+IMAX                                                           TDR2315
     XH=-XD(K)+XALP(K)*XC(M)*ERAD(M+1)+XA(K)*XGAM(K-1)*ERAD(N-1)+(CMXK-TDR2316
    11.)*XB(K)/CMXK*(XGAM(K)*ERAD(N)+XC(K)*ERAD(K+1))                   TDR2317
     XG(K)=(XH-XALP(K)*XG(M)-XA(K)*XG(K-1))/XB(K)                       TDR2318
510  CONTINUE                                                           TDR2319

     BACKWARD PASS -- EQ. 52 OF OLIPHANT                                TDR2320

     ERADM = 0.
     ERRORM = 0.
     DO 530 L=2,KMAX                                                    TDR2323
     K=KMAXA+1-L                                                        TDR2324
     N=K+IMAX                                                           TDR2325
     ENEW=CMXOM*(XG(K)-XGAM(K)*ERAD(N)-XC(K)*ERAD(K+1))+(1.-CMXOM)*ERADTDR2326
    1(K)                                                                TDR2327
     ERADM = AMAX1(ERADM,ENEW)
     ERRORM = AMAX1(ERRORM,ABS(ERAD(K) - ENEW) )
520  ERAD(K)=ENEW                                                       TDR2335
530  CONTINUE                                                           TDR2336
```

Modifications to Subroutine TDRAD (ADI) for SOR

```
      FOR SUCCESSIVE OVER RELAXATION

      REPLACE TDR2054 - TDR2055 BY


      TO RUN IN HECTIC A WAY MUST BE PROVIDED TO STORE THE
           EF ARRAY ON A DRUM ----- TOO LARGE TO HOLD IN CORE.


      DIMENSION IL(50),IR(50),EF(15000),RES(1200),EP(1)
      DIMENSION LBASE(50)
      EQUIVALENCE (X5,EP)
      INTEGER  R,RMAX,RXM1
      INTEGER RXIW

      A WAY MUST BE FOUND TO CALCULATE THE PANEL BOUNDERIES

      DATA IL/1,4,7,10,13,16,44*0/, IR/5,8,11,14,17,20,44*0/, LMAX/6/
      DATA XLAM/1.3/

      REPLACE TDR2278 - TDR2279 BY

  420 XB(K) = XB(K) + 3.E10*PLANCK(K) - XA(K) - XC(K) - XALP(K)         TDR2276
     1              - XGAM(K)
      XD(K)=XD(K)-4.104E12*PLANCK(K)*B(K)                               TDR2279

      REPLACE TDR2294 - TDR2336 BY


      START RELAXATION SOLUTION

      INITIALIZE

  450 ERADM = 0.
      ERRORM = 0.
      LBASE(1) = 0

      START LOOP ON L

      DO 480 L=1,LMAX
      IRL = IR(L)
      IRL = IR(L)
      ILL = IL(L)
      IW = IRL - ILL + 1
      IWP1 = IW + 1
      IWP2 = IW + 2
      K1 = IMAX - IW
      KFI = IW + IWP1
      KFM1 = KFI - 1
      RMAX = IW*JMAX
      RXM1 = RMAX - 1
      RXIW = (RMAX + 2)*IWP1
      KFMAX = RMAX*KFI - IW
      LB = LBASE(L)
      LBASE(L+1) = LB + KFMAX + IW
```

```
      IF(ITER.GT.1) GC TO 461

      LOAD EF ARRAY

      DO 452 KF=1,KFMAX
      EF(LB+KF) = 0.
  452 CONTINUE
      K = ILL - KI
      KF = LB -IW
      DO 458 J=1,JMAX
      K = K + KI
      DO 458 I=ILL,IRL
      K = K + 1
      KF = KF + KFI
      EF(KF-IW) = XALP(K)
      EF(KF) = XB(K)
      EF(KF+IW) = XGAM(K)
      IF(I.EQ.ILL) GO TO 454
      EF(KF-1) = XA(K)
      IF(I.NE.IRL) GO TO 456
      EF(KF+1) = 0.
      GO TO 458
  454 EF(KF-1) = 0.
  456 EF(KF+1) = XC(K)
  458 CONTINUE

      DO ELIMINATION ON EF ARRAY

      KF = LB - IW
      DO 460 R=1,RXM1
      KF = KF + KFI
      KFN = KF
      NX = MIN0(IW,RMAX-R)
      DO 460 N=1,NX
      KFN = KFN + KFNI
      IF(EF(KF).EQ.0.) STOP
      EF(KFN) = - EF(KFN)/EF(KF)
      FACT = EF(KFN)
      IF(FACT.EQ.0.) GO TO 460
      DO 459 M=1,NX
      EF(KFN+M) = EF(KFN+M) + FACT*EF(KF+M)
  459 CONTINUE
  460 CONTINUE

      COMPUTE RESIDUALS

  461 K = ILL - KI
      R = 0
      DO 462 J=1,JMAX
      K = K + KI
      DO 462 I=ILL,IRL
      R = R + 1
      K = K + 1
      RES(R) = - XD(K) - XALP(K)*ERAD(K-IMAX) - XGAM(K)*ERAD(K+IMAX)
     1          - XA(K)*ERAD(K-1) - XB(K)*ERAD(K) - XC(K)*ERAD(K+1)
```

```
462 CONTINUE

    DO ELIMINATION ON RESIDUALS

    KFN = LB - IW
    DO 464 R=1,RXM1
    NX = MIN0(IW,RMAX-R)
    KFN = KFN + KFI
    DO 463 N=1,NX
    KFN = KFN + KFNI
    RES(R+N) = RES(R+N) + RES(R)*EF(KFN)
463 CONTINUE
    KFN = KFN - KFNI*NX
464 CONTINUE

    DO BACK SUBSTITUTION

    KF = LB +KFMAX + KFI
    DO 468 R=1,RMAX
    KF = KF - KFI
    KR = RMAX - R + 1
    NX = MIN0(IW,RMAX-KR)
    IF(NX.EQ.0) GO TO 467
    DO 466 N=1,NX
    KFN = KF + N
    RES(KR) = RES(KR) - EP(KR+N)*EF(KFN)
466 CONTINUE
467 EP(KR) = RES(KR)/EF(KF)
468 CONTINUE

    INCREMENT ERAD

    R = 0
    K = ILL - KI
    DO 470 J=1,JMAX
    K = K + KI
    DO 470 I=ILL,IRL
    R = R + 1
    K = K + 1
    ERADP = ERAD(K)
    ERAD(K) = ERAD(K) + XLAM*EP(R)
    ERRORM = AMAX1(ERRORM,ABS(ERAD(K) - ERADP))
    ERADM = AMAX1(ERADM,ERAD(K))
470 CONTINUE
480 CONTINUE

    END LOOP ON L
```

Subroutine SPEW

```
     PRINTS 2-D ARRAYS(I,J)
          1,JMAX    ...    IMAX,JMAX
            .                   .
            .                   .
            .                   .
          1,1       ...    IMAX,1

      SUBROUTINE SPEW(ARRAY,IMAX,JMAX,BCD)
      DIMENSION ARRAY(1)
      IL = 1
      IR= MIN0(IMAX,10)
      JT = JMAX
      JB = MAX0(JMAX-49,1)
   10 PRINT 1000,  BCD
   15 PRINT 1010, (I,I=IL,IR)
      DO 20 JJ=JB,JT
      J = JT - JJ + JB
      K1 = IMAX*(J-1) + IL + 1
      K2 = IMAX*(J-1) + IR + 1
      PRINT 1020, J,(ARRAY(K),K=K1,K2)
   20 CONTINUE
      IF(JB.EQ.1) GO TO 30
      JT = JB - 1
      JB = MAX0(JT-49,1)
      GO TO 10
   30 IF(IR.EQ.IMAX) GO TO 40
      IL = IR + 1
      IR = MIN0(IMAX,IR+10)
      JT = JMAX
      JB = MAX0(JMAX-49,1)
      IF(JMAX.GT.20.OR.IMAX.GT.40) GO TO 10
      PRINT 1030, BCD
      GO TO 15
   40 RETURN
 1000 FORMAT(1H1,A6)
 1010 FORMAT(1H0,6X,10I12)
 1020 FORMAT(1H ,I6,10E12.5)
 1030 FORMAT(1H0,A6)
      END
```

REFERENCES

1. Lax, P. D., and B. Wendroff, "Systems of Conservation Laws," Comm. Pure Appl. Math. 13, 217 (1960).
2. Richtmyer, R. D. and K. W. Morton, Difference Methods for Initial Value Problems, 2d ed., Interscience Publishers, New York, 1967.
3. Burstein, Samuel Z., "Finite-Difference Calculations for Hydrodynamic Flows Containing Discontinuities," J. Computational Phys. 2, 198 (1967).
4. "Nuclear Explosion Interaction Studies," v. I, "Methods for Analysis of Radiative Transfer," Air Force Weapons Laboratory Report AFWL-TR-66-108, General Dynamics, General Atomic Division, Sections III and IV, to be issued.
5. Freeman, B. E., and C. G. Davis, "Fireball Phenomenology and Code Development," v. III, "SPUTTER Subroutines for Radiation Transport in Spheres," Air Force Weapons Laboratory Report AFWL-TR-65-143, General Dynamics, General Atomic Division, August 1965.
6. Carlson, B., "Numerical Solution of Neutron Transport Problems," Proc. Symp. Appl. Math. XI, 219 (1961).
7. Leiner, A. L. et al, "PILOT - A New Multiple Computer System," Assoc. Comp. Mach. 6, 315 (1963).
8. Freeman, B. E., "The Non-Equilibrium Diffusion Equation," USAEC Report LA-3377, Los Alamos Scientific Laboratory, 1965.
9. Oliphant, T. A., "An Extrapolation Procedure for Solving Linear Systems of Equations," Quart. Appl. Math. 20, 257 (1962).
10. Wachpress, E. L., "Iterative Solution of Elliptic Systems and Applications to the Neutron Diffusion Equations of Reactor Physics," Prentice-Hall, Englewood Cliffs, New Jersey, 1966.

11. Peaceman, D. W., and H. H. Rachford, Jr., "The Numerical Solution of Parabolic and Elliptic Differential Equations," J. Soc. Ind. Appl. Math. 3, 28 (1955).

12. Southwell, R. V., Relaxation Methods in Theoretical Physics, The Clarendon Press, Oxford, 1946.

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Gulf General Atomic Incorporated San Diego, California 92112	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

NUCLEAR EXPLOSION INTERACTION STUDIES, Volume II, Two-Dimensional Code Development

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

29 September 1966-27 October 1967

5. AUTHOR(S) (First name, middle initial, last name)

J. R. Triplett et al.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
April 1968	332	12

8a. CONTRACT OR GRANT NO. F29601-67-C-0014 b. PROJECT NO. 5710 c. Subtask No. 07.017 ARPA Order No. 313 d.	9a. ORIGINATOR'S REPORT NUMBER(S) AFWL-TR-67-131, Vol II 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) Contractor's report No. GA-7764, Vol II

10. DISTRIBUTION STATEMENT This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of AFWL (WLRT), Kirtland AFB, NM, 87117. Distribution is limited because of the technology discussed in the report.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	AFWL (WLRT) Kirtland AFB, NM 87117

13. ABSTRACT

(Distribution Limitation Statement No. 2)

The HECTIC code is a two-dimensional Eulerian radiative hydrodynamics code designed for nuclear and laser phenomenology applications involving primarily vapor-phase materials, with provisions for heat conduction and vaporization of condensed matter. In this report, a formulation is presented for an improved method of solving the hydrodynamic equations. Solution of the radiative transfer equations by the long-characteristic method is discussed, and computer codes utilizing this approach are presented. The short-characteristic method is also discussed. The nonequilibrium diffusion approximation in two dimensions is considered, and a study of algorithms for solving the implicit difference equations which arise is reported. Experimental codes utilizing the nonequilibrium diffusion and short-characteristic methods are presented.

DD FORM 1 NOV 65 1473

UNCLASSIFIED
Security Classification

UNCLASSIFIED
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Radiative hydrodynamics codes						
Two-dimensional radiative transfer codes						
Two-dimensional hydrodynamics equations						

AFSC (KAFB NM)

UNCLASSIFIED
Security Classification